FACULTEIT ECONOMIE
EN BEDRIJFSWETENSCHAPPEN

KU LEUVEN

PERSONNEL PLANNING IN SERVICE SYSTEMS WITH NONSTATIONARY DEMAND

Proefschrift voorgedragen tot
het behalen van de graad van
Doctor in de Toegepaste
Economische Wetenschappen

door

Mieke DEFRAEYE

Nummer 439 2014

Doctoral Committee

Advisor: Prof. dr. Inneke Van Nieuwenhuyse
KU Leuven

Members: Prof. dr. Marc Lambrecht
KU Leuven

Prof. dr. Nico Vandaele
KU Leuven

Dr. Galit Yom-Tov
Technion - Israel Institute of Technology

Dr. David Worthington
Lancaster University

Prof. dr. Jeroen Beliën
KU Leuven @HUB

Acknowledgement

And so it begins... the section of this dissertation that will undoubtedly be read by the most people. Not without reason though, because these three pages with expressions of gratitude towards colleagues, friends, and family are at least as important as what will follow in the remainder of this dissertation.

First of all, I would like to sincerely thank my supervisor, Inneke Van Nieuwenhuyse. In 2009, she introduced me to the topic of scheduling with time-dependent demand and immediately triggered my curiosity by outlining a rather ambitious research project, with many challenges and uncertain outcomes. Over the years, I have come to know Inneke as a dedicated supervisor, who spends *a lot* of time and effort into the guidance of her PhD students. What I appreciate most about Inneke, perhaps, are her people skills: she always communicates in a very straightforward, honest, and open way, and guides her PhDs with the best possible intentions. Inneke's professional guidance and invaluable support have contributed significantly to the quality of this dissertation.

During my PhD, Inneke has given me the opportunity to attend a number of conferences, which I have always found inspiring (from a professional point of view) and relaxing (because for some reason, being abroad always feels like a holiday to me). The advantage of attending conferences is that you run across people who share the same research interests; this was how I met Galit Yom-Tov and David Worthington, who are now members of my Doctoral Committee. Galit always had a fresh view on my research. She read my work thoroughly and provided me with detailed comments; I'd like to thank her for that. I came to know David as a very intelligent and mod-

est person, with a genuine curiosity and enthusiasm about research. I wish to thank him for sharing his thoughts on my work and for the interesting discussions we had.

I would like to extend my thanks to Marc Lambrecht and Nico Vandaele. As the co-supervisors of my PhD project, Marc and Nico carefully questioned my research from a practice-oriented point of view. They regularly confronted me with questions that were (to a greater or lesser extent) outside my comfort zone, encouraging me to think differently and to position my research in a broader context. Their remarks definitely made my PhD much richer, and I am grateful for this.

Finally, I thank Jeroen Beliën, who was the last one to join my Doctoral Committee. In my opinion, Jeroen is somehow "a special case": like no other, he succeeds in combining high-level research and ambition with a more than average dose of self-mockery, enthusiasm, and optimism – I admire this. Inneke, Galit, David, Marc, Nico, and Jeroen: thank you for the interesting discussions, your remarks, and your support. They allowed me to improve this dissertation considerably.

In addition to my supervisor and the members of the Doctoral Committee, I would like to thank Erik Demeulemeester, whose advice has been of great help in improving some of my work. I also wish to thank Jorne, Patricio, Jeroen (again), and Stefan. They have some features in common, which a greatly appreciate: they are clever (not only the solution methods they develop), they are talented academics, and they all have a truly great sense of humor. I very much enjoyed the conversations about Research (and who would be/could become the God of the HOG), and all other experiences we shared. Though our big ideas have not found their way to the academic journals yet, I certainly hope we can realize this in the future. Special thanks go to Stefan – for many reasons, among others for the discussions we had while the running in the forest, that brought forward an endless list of topics for future research.

Finally, I am thankful to the Research Foundation-Flanders (FWO), for giving me the opportunity and funding to pursue a PhD (and in the same line, my brother Bert, who indirectly contributed to my PhD funding).

This brings me to the colleagues. It goes without saying that the work atmosphere is what makes our group so special (and I now speak for both P&L and our friends at the 5^{th} floor). It has been a pleasure to meet you all, and I feel very lucky to work in such a lively group that successfully combines the serious work-related stuff with good, genuine fun. We spent a lot of after-work time on team building and bonding, and I can confidently conclude that we achieved our goal (the team was built, indeed!). In addition, the sensible and nonsensical conversations we had during the breaks always were a very effective way to recharge my batteries, and the spontaneous pizza evenings with Patricio, Morteza, Ruth and Carla were just what I needed to keep me going during the busy/busier periods of my PhD. Lastly, a very special word of thank goes to Yannick (my officemate for more than 4 years) and Ann (my officemate next door). With Yannick –who is most probably the best officemate ever– I shared the joys and frustrations well-known to every PhD student, as well as a common interest in chocolate (in a slightly different way, though).

Many thanks go to my dearest friends and their families, for making sure that there is enough "life" in my work-life balance. Special thanks goes to the Stam at Scouts Egenhoven: I very much enjoyed the activities we organized. They were sometimes distracting, time-consuming, or maybe dangerous, but always hilarious.

A final word of thanks goes to my parents and brothers, who –without question– excel in doing all the things a family should do. My sincerest gratitude goes to my parents: the energy, dedication and enthusiasm they put into any plan or project they set their minds on, is very inspiring.

Thank you all!

Mieke Defraeye
Leuven, January 2014

Abstract

In many service systems, the number of arriving customers is not constant over time: the number of customers fluctuates over the course of the day, week, month, or year according to a stochastic (though to some extent predictable) pattern. Such time-varying arrival rates are frequently observed not only in emergency departments (EDs), but also in call centers, banks, and retail stores. If the personnel capacity is not properly adjusted to this time-varying demand for service, customer waiting times may increase severely. This particularly holds for healthcare settings such as the ED, where long waiting times are particularly undesirable because a patient's condition may severely worsen while waiting for treatment. Adequate personnel capacity planning is often the key tool to prevent the fluctuations in the demand for service to inflate the customer waiting time.

In this dissertation, we study how customer quality of service (waiting times, in our case) can be controlled in systems with a nonstationary arrival process (i.e., time-dependent and stochastic arrivals), by selecting the appropriate staffing levels and/or shift schedules throughout the day. We contribute on two aspects of personnel scheduling with nonstationary arrival rates: *performance measurement* (i.e., how to measure the time-varying customer waiting time, for a given staff schedule) and *workforce optimization* (i.e., determining the lowest cost staff schedule that meets the target performance with respect to customer waiting times).

This thesis consists of three main parts: in the first part (Chapter 2), we present an extensive literature review on personnel staffing and scheduling in systems with nonstationary arrival rates. We categorize the available articles based on classification criteria such as the system assumptions, per-

formance evaluation characteristics, optimization approaches and real-life application context. We discuss the main challenges and outline opportunities for continued research.

In the second part of this thesis (Chapter 3), we study how to evaluate the time-dependent customer quality of service pertaining to any given staffing or scheduling solution (i.e., performance measurement). This is highly important when making capacity decisions, because capacity planning models require a performance measurement method as a subroutine, to assess the solution quality of any given plan. We compare several methods to compute time-dependent waiting times in small-scale service systems, with a particular focus on methods that are capable of addressing the following (realistic) problem features: nonstationary arrivals, general service and abandonment time distributions, and an exhaustive service policy. Measuring waiting times in systems with nonstationary arrivals is not trivial: the traditional stationary queueing models might no longer be applicable, because the arrival rate fluctuates over the day. Moreover, the existing models that are intended for systems with nonstationary arrivals often rely on rather theoretical assumptions (such as exponential service and abandonment times).

In the third part of this thesis, we present two solution methods for workforce optimization in service systems with nonstationary demand (Chapters 4 and 5). The first method is a simulation-based heuristic that solves the staffing problem: we determine the time-varying staffing levels that need to be available to achieve a target service level on the customer waiting time (shift constraints are disregarded). Our heuristic succeeds in consistently providing good solutions and does not require strong theoretical assumptions (in contrast to many other heuristics in the literature). The second method extends the scope by including shift constraints in the analysis: we present a simulation-based branch-and-bound approach for shift scheduling with time-dependent arrival rates. The method is best suited for small-scale systems with limited opening hours.

Samenvatting

In de dienstensector fluctueert het aantal klanten dat aankomt meestal over een dag, week, maand of jaar volgens een stochastisch (maar enigszins voorspelbaar) patroon. Als de personeelscapaciteit onvoldoende afgestemd is op deze tijdsafhankelijke vraag naar diensten, kan de wachttijd van klanten hier ernstig onder lijden. Dit geldt vooral voor organisaties in de gezondheidszorg, zoals spoedafdelingen. Lange wachttijden zijn daar zeer onwenselijk, omdat de toestand van een patiënt ernstig kan verergeren tijdens het wachten op behandeling. Deze tijdsafhankelijke aankomstritmes komen niet enkel voor in spoedafdelingen, maar ook in call centers, banken, en winkels. Personeelsplanning is dan vaak een zeer belangrijk instrument om te vermijden dat de schommelingen in het aankomstritme zich vertalen in hogere wachttijden voor de klant.

In dit proefschrift bestuderen we hoe de kwaliteit van dienstverlening (wachttijd, in ons geval) onder controle kan gehouden worden in dienstensystemen met niet-stationaire aankomstritmes (d.w.z., met een tijdsafhankelijk en stochastisch aankomstproces). Dit doen we door na te gaan hoeveel personneel er nodig is op elk moment van de dag en vervolgens de shiftplanning te bepalen. Ons onderzoek draagt bij aan de literatuur rond personeelsplanning in systemen met een niet-stationair aankomstproces op twee gebieden: performantiemeting (i.e., het meten van tijdsafhankelijke wachttijden, voor een gegeven werkshift) en optimalisatie van de personeelsplanning (d.w.z., het bepalen van een shiftplanning die voldoet aan de vooropgestelde wachttijden, aan lage kost).

Dit proefschrift bestaat uit drie grote delen: in het eerste deel (Hoofdstuk 2) presenteren we een uitgebreide literatuurstudie over personeels-

planning in systemen met een niet-stationair aankomstproces. We categoriseren de bestaande literatuur op basis van vier classificatiecriteria: de systeemassumpties, de performantiemaatstaven, de optimalisatiemethodologie, en het vooropgestelde toepassingsgebied. We bespreken de belangrijkste uitdagingen in het onderzoeksgebied en formuleren topics die verder onderzoek vereisen.

In het tweede deel van dit proefschrift (Hoofdstuk 3) bestuderen we hoe het serviceniveau gevalueerd kan worden doorheen de tijd, voor een gegeven shiftplan. Dit proces is zeer belangrijk met het oog op het optimaliseren van de personeelsplanning: performantie-evaluatiemethodes vormen een onmisbare component in optimalisatiemodellen, omdat ze toelaten de wachttijd van een bepaald shiftplan te evalueren. We vergelijken verschillende methodes die toelaten de (tijdsafhankelijke) wachttijd te berekenen in kleinschalige dienstensystemen met een tijdsafhankelijk aankomstenpatroon. We focussen in het bijzonder op methodes die in staat zijn om meer realistische eigenschappen te modelleren, zoals o.a. een niet-stationair aankomstenproces, algemene (statistische) verdelingen voor de procestijd en het geduld van klanten, en de mogelijkheid om overtijd te werken op het einde van een shift. Het meten van wachttijden in systemen met een niet-stationair aankomstenproces is niet evident: de traditionele (stationaire) wachtlijnmodellen kunnen vaak niet meer toegepast worden, net omdat de aankomsten fluctueren doorheen de dag. Bovendien zijn de bestaande modellen voor systemen met niet-stationaire aankomsten vaak gebaseerd op een reeks beperkende, theoretische assumpties.

In het derde deel van dit proefschrift introduceren we twee oplossingsmethodes voor het optimaliseren van de personeels- en shiftplanning in systemen met een niet-stationaire vraag naar diensten (Hoofdstukken 4 en 5). De eerste methode is een heuristiek, die gebruik maakt van simulatie. We lossen het zogenaamde "staffing probleem" op, dat de tijdsafhankelijke capaciteit bepaalt die nodig is om een vooropgesteld serviceniveau te halen m.b.t. de wachttijden (beperkingen gerelateerd aan de werkshiften worden nog niet in rekening gebracht). In tegenstelling tot vele andere heuristieken in de literatuur, biedt onze methode consistent goede oplossingen, en dit zonder sterk beperkende assumpties te maken. De tweede methode is ruimer van

insteek en neemt ook beperkingen op in verband met de shiften (zoals shift-duur en starttijdstip): we stellen een simulatie-gebaseerd branch-and-bound algoritme voor, dat vooral mikt op kleinschalige systemen met beperkte openingstijden.

Contents

Chapter 1

Introduction

In many service systems, the number of arriving customers varies with a daily, weekly or monthly recurring pattern. If the personnel capacity is not adjusted accordingly, these variations in the arrival rate may inflate customer waiting times substantially. This issue is particularly relevant in healthcare settings such as the emergency department (ED), where excessive waiting times are highly undesirable. Adequate personnel capacity planning then is crucial, because the ED cannot resort to waiting lists or appointment systems to mitigate fluctuations in the customer arrival rate [128]. More generally, capacity planning is challenging in service organizations, because (1) services are direct (they cannot be inventoried), (2) they require interaction between the provider and the consumer [212], and (3) the variability in the demand for service is high (i.e., fluctuating arrival rates are common; [1, 128]).

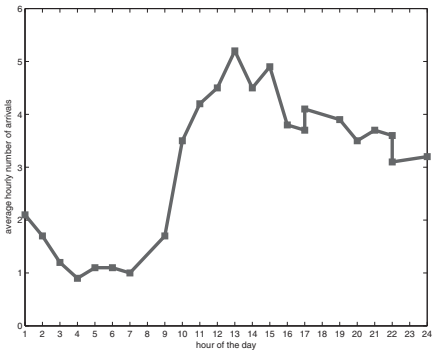
In this dissertation, we study how customer quality of service can be controlled in service systems with a time-dependent stochastic (i.e., nonstationary) arrival process, by adjusting the personnel capacity to the variations in the arrival rate of customers into the system. The overall objective is to effectively manage the trade-off between labor cost and customer quality of service (waiting times, in our case). To prevent the customer quality of service to fluctuate heavily over the course of a day (as a result of the time-dependent arrival rates), the personnel capacity is altered to better match the demand for service.

In Section 1.1, we provide illustrations of the systems upon which we focus in this thesis and we highlight the relevance of our research. Section 1.2 then proceeds with a broad-scope discussion of service strategies that can be applied to cope with the time-dependent demand for service. We elaborate on capacity management (Section 1.3) and demand management (Section 1.4) in the next sections, mainly from the viewpoint of the environments we target (EDs, call centers, banks, retail stores). In Section 1.5, we define the scope of this dissertation and provide an outline of what follows in the next chapters.

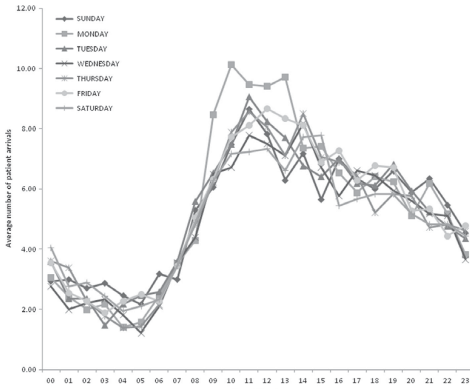
1.1 Research relevance

Although the exact arrival time of individual customers is usually not known in advance, a daily recurring pattern can often be distinguished. Figure 1.1 shows the expected number of arrivals to EDs in New York (US, [99]), Ontario (Canada, [163]), and an ED in a regional hospital in Belgium [65]. The similarities are apparent: all EDs are clearly confronted with peak demand in the morning hours and early afternoon; demand then stays relatively high until 8PM, after which it declines rather steeply during the evening hours and stays low during the night. This illustrates that the variability in demand often is predictable to a certain degree. In the ED, mismatches between capacity and the demand for service have a major impact on the quality of service: excessive waiting may severely worsen a patient's condition [195] or cause the patient to leave without receiving treatment [134]. Because the ED cannot resort to waiting lists or appointment systems to mitigate fluctuations in customer arrival, capacity planning is the main tool to keep waiting times under control. Other emergency services, such as ambulance scheduling [241] and police patrol scheduling [150, 67], face comparable challenges.

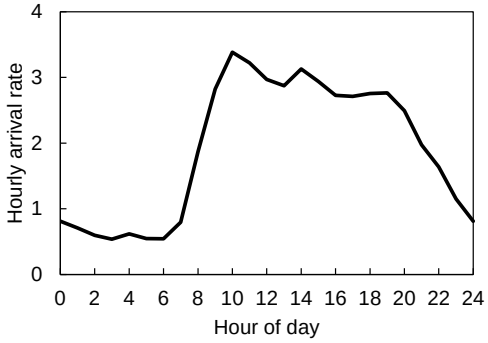
These recurring, predictable arrival rate patterns are omnipresent not only in emergency services, but also in a call center workforce scheduling [85, 7, 181], personnel scheduling in restaurants or banks [115], cashier scheduling in retail stores [139, 157], and scheduling personnel at check-in counters and at customs service in airports [156, 165, 206, 196] (Figure



(a) ED in New York (source: Green et al. [99])

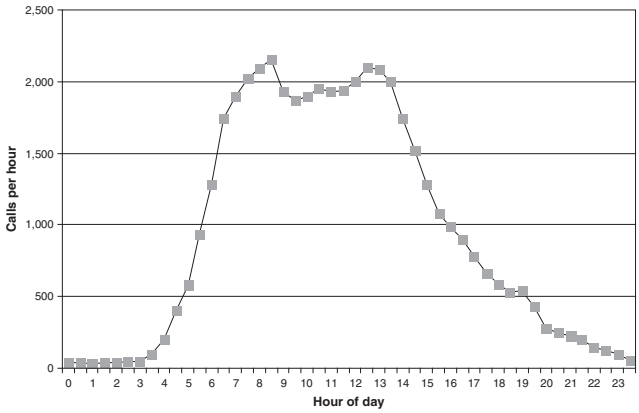


(b) ED in Ontario (source: Lim et al. [163])

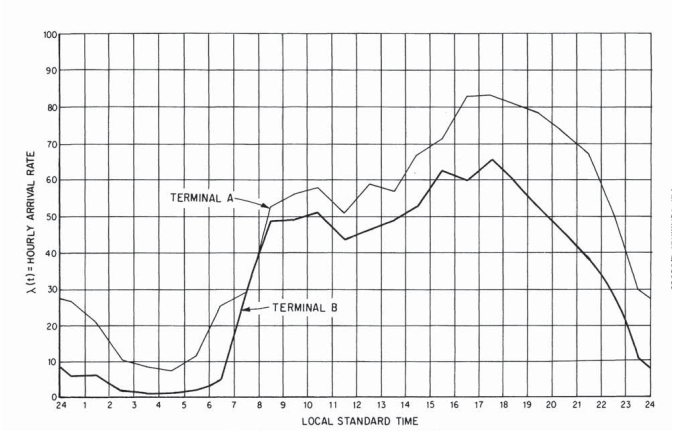


(c) ED in a regional hospital in Belgium (source: Defraeye and Van Nieuwenhuyse [65])

Figure 1.1: Illustrations of ED arrival rate patterns



(a) Call center arrivals (source: Feldman et al. [78])



(b) Air terminal arrivals (source: Koopman [156])

Figure 1.2: Illustrations of time-varying arrival rates in the call center and airline industry

1.2 provides examples from a call center and an airport terminal). Here, waiting times are usually controlled in view of maximizing profits: long queues and waits may cause customers to abandon, resulting in lost sales and/or lost customers. Dean [63] assert that the service quality has a major influence on customer loyalty and retention, in call centers. Netessine et al. [194] conclude that planning labor based on store traffic will improve a retail store's sales: their analysis indicated that a 3% sales increase can be realized with only modest improvements in the employee schedule (and how it is executed). The predictability of the demand is perhaps even most outspoken in airport settings, where the distribution of the customer arrival time (relative to the flight departure time) can be measured as a function of the time of day, through so-called check-in curves [206]. Combining this information with the flight arrival and departure times, managers have a clear indication of when the arrival peaks at check-in counters and customs service will occur, and can staff accordingly.

The settings described above have the following characteristics in common; these features define the environments we target in this dissertation:

- they are all service systems,
- with a time-varying stochastic arrival process (that is to some extent predictable),
- where controlling customer waiting times is important,
- and personnel capacity is the main tool to do so.

Balancing capacity and quality of service is an important challenge in the service sector [216], and the peaks in demand make this trade-off even more challenging. However, despite the practical relevance and the increasing attention in the academic literature [99, 253], fluctuations in the arrival rate often do not receive sufficient attention in real-life personnel capacity plans [100, 21, 218]. Decision makers continue struggling to adjust capacity planning to account for the fluctuations in customer demand [30] and as such, operational models that provide a systematic approach to capacity planning are of great value.

1.2 Strategies to cope with nonstationary arrivals

Nonstationary arrival processes are stochastic processes (i.e., the exact arrival times are unknown) for which the parameters of the arrival distribution fluctuate over time (cf. Figure 1.3), thus creating periods of peak demand (or busy periods) and periods of low demand.

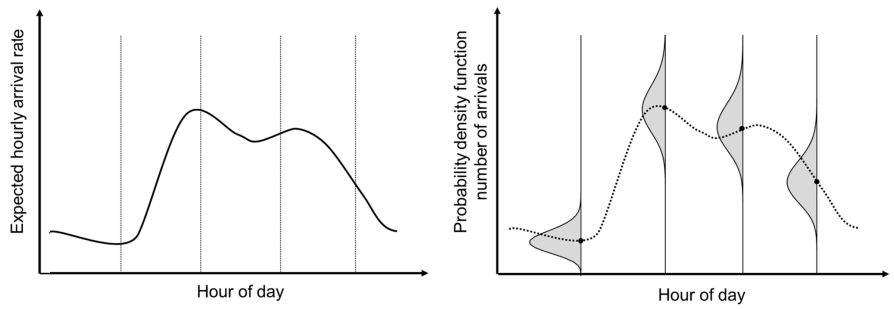


Figure 1.3: Nonstationary arrival process

Sill [217] present the following three strategies to cope with periods of peak demand:

- Adjust capacity to account for the fluctuations in the demand for service
- Manipulate demand to match the available capacity
- Let customers wait

This distinction relates to two well-known service management strategies, which date back to Sasser [212]: level-capacity and chase-demand. The *level-capacity* strategy keeps capacity at a constant level. If the demand fluctuates, this may induce long customer waiting times (especially during busy periods). Green et al. [95] explore to what extent the nonstationarity in the arrival process affects system performance (in a system with constant capacity): the expected waiting time, the probability that a customer has to wait and the queue size all increase as the system becomes more nonstationary. Evidently, the impact is more pronounced as the arrival rate fluctuations grow larger in size. The *chase-demand* strategy, on the other hand, uses

flexible capacity to account for the fluctuations in the arrival rate. However, this strategy can only be applied if altering capacity is feasible in sufficiently short time [12].

Many researchers have extended and complemented these basic strategies; an insightful overview can be found in Klassen and Rohleder [147]. Heskett et al. [112] discuss intermediate strategies that apply when strict chase-demand nor level-capacity is appropriate. They are classified based on the predictability of demand and the possibility to shift customer demand. For instance, if demand can be predicted but customer demand cannot be shifted easily (as is the case in the environments we study), then capacity planning is the preferred tool to account for the fluctuations in demand [112, 147]. Armistead and Clarck [12] put forward another intermediate strategy (the *coping strategy*), which strives to minimize the drop in customer service that follows after a capacity shortage. At certain times, the capacity will inevitably be insufficient to satisfy the demand for service within the target time frame, the system then enters the so-called "coping-zone". Organizations can deal with periods of "coping" by, for example, focussing on problematic customers, providing action/escalation teams that become active during the period of coping, by simplifying or shortening the service process during the periods of capacity shortage, or by providing a more basic service during busy periods [12].

The appropriateness of any strategy depends on the service system under study: banks, retail stores, emergency departments and restaurants have limited control over the demand side, which thus increases the importance of capacity planning. The degree to which customer waiting can be used as a tool to cope with nonstationary demand depends on the importance of delivering consistent customer service [12], and therefore is dictated by the problem setting: emergency services tend towards a chase-demand strategy, because customer waits should remain limited. However, if long waits are acceptable, capacity does not necessarily need to be in line with demand peaks and strategies similar to the level-capacity strategy may be more suitable.

In this dissertation, the trade-off between customer quality of service and personnel capacity planning is central. As such, our research adheres closely to the chase-demand strategy.

1.3 Managing capacity

Capacity management can be defined as "*the ability to balance demand from customers and the capability of the service delivery system to satisfy the demand*" [12]. A broad range of capacity management options (CMOs) is listed in Klassen and Rohleder [147, 148], they are classified as "must do" versus optional, and short-term versus long-term (we present a selection in Table 1.1). In this section, we place more emphasis on *personnel* capacity management because labor is crucial to guarantee high service quality [127]. However, we acknowledge that capacity includes personnel as well as other resources (e.g., machines, equipment, hospital beds, facilities) and that both need to be managed effectively. For further details we refer to Klassen and Rohleder [147, 148]; a recent in-depth overview of capacity management strategies can be found in Pullman and Rodgers [199].

The table reveals that employee scheduling is of crucial importance in service organizations. Optional short-term capacity planning options are to allow employees to work overtime (if capacity is insufficient), to tolerate idle time (in case of temporal overcapacity), or to acknowledge that customers will have to wait during busy periods. Though idle time is expensive in terms of labor cost, a certain amount of overstaffing may be a valid option in settings where customer waiting should be avoided (e.g., in emergency services). In a similar manner, allowing employees to differentiate between services that can and cannot wait is beneficial: postponing non-urgent jobs (such as administrative tasks or answering e-mails) to low-demand periods reduces the pressure during periods of peak demand [212, 256]. This approach has been applied for call center staffing in Liao et al. [161]. On the longer term, hiring and layoff decisions, and yearly employee scheduling are most important. Cross-training employees is particularly appealing in settings where different types of services are offered: the employees can then switch to whatever service experiences a capacity shortage. This is common practice in call centers [85, 7]; we refer to Aksin et al. [8] for a review on cross-training in call centers. This practice appears to be less common in healthcare, though the use of cross-trained nurse pools is increasing in importance [40, 89, 116]. Cross-trained staff can be notably more expensive,

	Shorter-term CMOs	Longer-term CMOs
Required CMOs (must do)	Employee scheduling (hourly, daily weekly)	Hiring employees Layoff employees Yearly scheduling (e.g., vacations)
Optional CMOs (may do)	Temporary employees Allow overtime Allow idle time Change work speed (employees work faster) Allow customer waiting Do non-urgent work in calm periods Turn away customers Allow customer balking	Part-time employees Rent capacity from other businesses Share capacity with other businesses Cross-train employees Simplify the service process Re-design service

Table 1.1: Capacity management options (CMOs)
(source: adapted from [147, 148])

but offers extra flexibility.

As this dissertation focuses on short-term personnel planning, it can be positioned in the upper left quartile of Table 1.1. Though personnel planning has received considerable attention in the operations research literature [73, 240], the nonstationarity in the arrival process —and in particular its impact on the quality of service— is frequently overlooked. Personnel planning is typically decomposed into different stages, they are illustrated in Figure 1.4: (1) demand forecasting, (2) staffing, (3) shift scheduling, (4) rostering. Short-term updating of schedules can be considered as an additional step. In what follows, we discuss each of these steps in further detail.

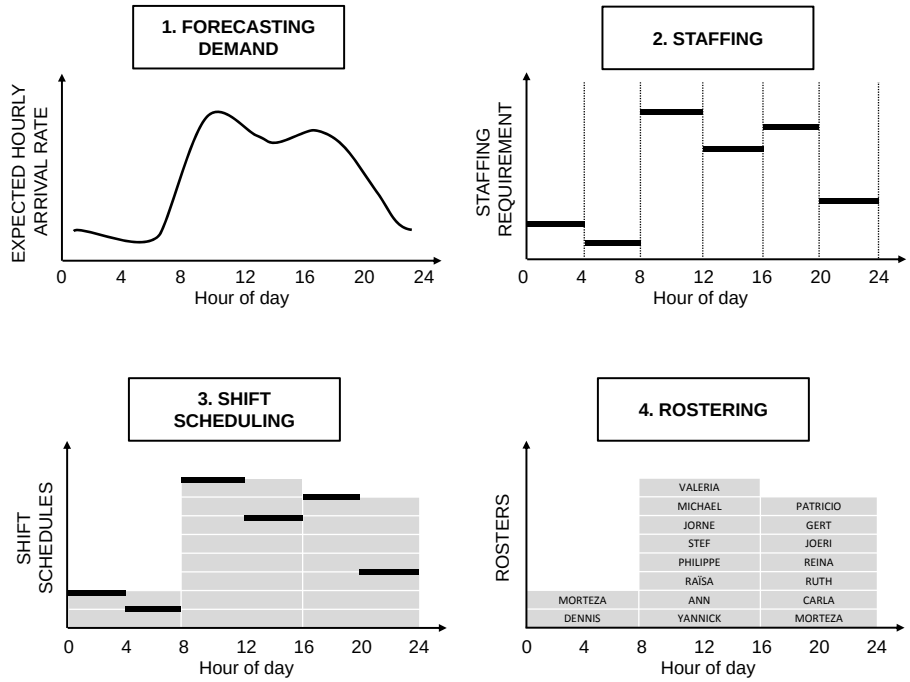


Figure 1.4: The four typical stages in personnel planning

Demand forecasting

First, estimates of the future arrival rates need to be derived from historical data. These demand forecasts serve as input to the subsequent steps

in the capacity planning process, so their accuracy is essential to obtain high-quality personnel schedules and rosters. As "intuitive" forecasts often underestimate the actual demand [135], systematic approaches are needed. Nevertheless, the current practice in forecasting arrival rates is often to apply crude and unsophisticated methods [241]. Gans et al. [85] describe arrival forecasting practice in call centers; literature reviews on demand forecasting can be found in Vile [241] and Wargon et al. [244] (the latter focuses on EDs). Details on forecasting methods can be found in Avramidis et al. [15], Brown et al. [38], Taylor [228], Shen and Huang [214], Millán-Ruiz and Hidalgo [191] (call centers); Jones et al. [136], Morzuch and Allen [192] (EDs); and Matteson et al. [186], Vile [241] (ambulance calls), among others.

Staffing

Next, the arrival patterns are converted into staffing requirements that meet the target quality of service. It is here that the trade-off between personnel capacity (and hence, labor cost) and quality of service is made. Various performance metrics can be used to describe the customer quality of service. Service-related measures are common, for instance in the context of staffing airline check-in counters (e.g., 85% of passengers wait less than 10 minutes; [196]), in call centers (e.g., Gans et al. [85] report waiting time targets of at most 20 or 30 seconds) and in emergency departments (e.g., triage systems that specify the maximum duration until the first examination by a doctor). Emergency departments tend to evaluate the total time spent in the ED (i.e., the length-of-stay): in the UK, a 4-hour target is used (see [126, 183], among others). Other relevant metrics include the probability of experiencing an excessive waiting time (i.e., the complement of the service level) and the probability of leaving without receiving treatment (i.e., the abandonment probability). Similar metrics are used in call centers, though the probability that a customer cannot be serviced immediately (i.e., the delay probability; [85]) is also common. The staffing process then searches for the number of personnel that is needed at any time of the day, to meet the target performance at low cost. This can be done by means of queueing models [99, 253] or simulation-optimization approaches [78]. Note that the staffing requirements may fluctuate heavily from one period to the next,

which implies that they are not necessarily feasible in practice.

Scheduling

As employees work according to shift types with a given start time and duration (e.g., an 8-hour shift that starts at 9AM, with a lunch break at noon), the staffing requirements need to be translated into a shift schedule. This schedule is constructed such that it (1) is in line with the staffing requirements, (2) is inexpensive in terms of labor cost, and (3) meets the shift constraints (e.g., shift duration, shift start time, work and meal breaks). The difference between the staffing requirements and the shift schedule (and their cost) depends on how stringent the shift constraints are. Constructing a shift schedule that closely follows the staffing requirements is easier if many different shifts types exist. The use of part-time employees ([114, 6], see also [240] for further references) or split shifts (i.e., a shift that consists of two distinct periods separated by a 2- to 4-hour break; [190, 18]) contribute greatly to the scheduling flexibility [26, 18, 19]. Atypical shift structures, however, tend to threaten safety and endanger employee health (e.g., fatigue-related accidents; [66, 33]).

Rostering

In a final phase, rosters are constructed by assigning individual workers to the schedule, while taking into account various constraints related to the allowed (or preferred) work patterns for individual workers [73]. In this respect, Van den Bergh et al. [240] distinguish between time-related constraints, and fairness and balance constraints[1]. Time-related constraints include official regulations (e.g., a limit on the number of hours an employee is allowed to work per day, week or month; limits on the minimum time between shifts); these are hard constraints – meeting them is mandatory. Soft time-related constraints are driven by employee preferences, such as the preferred the minimum number of consecutive working or non-working shifts. Fairness and balance constraints aim at limiting the differences among em-

[1]We assume that the coverage constraints, that dictate the number of staff needed to cover the workload, are met during the staffing phase.

ployees, with regard to the schedule that is assigned to each of them. Examples include an equal distribution of morning, evening and night shifts among all employees (and their timing throughout the year), and fairness in granting employee requests (such as preferences with regard to work location or preferences to work certain shift types). We refer to Ernst et al. [73] and Van den Bergh et al. [240] for an extensive discussion of the various types of rostering constraints.

Real-time schedule adjustments

An additional phase, which has received fairly little attention in the academic literature, is real-time updating of schedules. In many service organizations, additional information may become available on short notice, e.g., changes in demand or unavailability of staff (such as absence of employees due to illness). Managers then have the opportunity to alter the staff schedule based on this additional information. Though experienced managers tend to succeed in making adequate schedule updates based on intuition [118], computer-based approaches are helpful to further improve the performance. Evidently, accurate forecasts are critical to the successful application of real-time rescheduling methods [118]. We refer to Thompson [235], Hur et al. [118], Testik et al. [229], and Mehrotra et al. [189] for methods that address real-time updating of work schedules.

In this dissertation, we focus on personnel planning in service systems with nonstationary demand, and explore how the customer quality of service can be improved by selecting the personnel capacity in line with the time-varying arrival pattern. The emphasis lies on staffing and scheduling.

1.4 Managing demand

Demand management covers a set of proactive approaches that attempt to manipulate the time-varying arrival rate itself. As such, it represents an important alternative (and complementary approach) to capacity management, which is reactive in nature. The importance of demand management in service organizations has been studied by Sasser [212], Armistead and Clarck

[12], Crandall and Markland [56], Klassen and Rohleder [147, 148, 149], Jack et al. [127], and Le [160], among others.

Crandall and Markland [56] suggest four basic strategies (see also Klassen and Rohleder [147]), that complement the work of Sasser [212], Heskett et al. [112], and Armistead and Clarck [12]:

- Match: match capacity to the demand for service (i.e., similar to chase-demand)
- Provide: match (constant) capacity to the maximum demand (i.e., similar to level-capacity)
- Control: control demand such that it remains at a constant level
- Influence: first influence the demand to reduce fluctuations, and then match the capacity to the resulting demand pattern

Whereas the Match and Provide strategies focus solely on the capacity side, Control and Influence also include demand management (i.e., a certain degree of demand smoothing). The Control strategy limits the demand in a rather strict manner (e.g., by using an appointment system); Influencing can be considered as a combined demand and capacity management strategy. In this dissertation, the trade-off between customer quality of service and personnel capacity planning is central and as such, our research can be positioned within the Match strategy of Crandall and Markland [56] (i.e., match capacity to the demand for service), which is both the most commonly applied and the most highly valued strategy among practitioners [56]. Although the survey of Crandall and Markland [56] indicates that Influence is not a common strategy, managers expressed a strong desire to use this strategy more often.

Table 1.2 summarizes a range of demand management options (DMOs; see Klassen and Rohleder [147, 148] for a more elaborate list and an in-depth discussion). Essentially, demand management aims at smoothing demand and/or achieving a better match with the available capacity. Demand management has played a key role in the banking industry in recent years, where simple actions, such as informing customers when service is less busy or referring them to alternative services, have proven to be effective [149]. Other notable examples of successful demand management initiatives in the bank

sector include the introduction of ATMs (i.e., automation), and more recently on-line banking (which increases customer participation). However, Klassen and Rohleder [149] emphasize that the potential to automate may be far less outspoken in other sectors (e.g., medical services which need to be performed performed "on the customer", or services that relate to customer experiences). Klassen and Rohleder [149] assert that customer flexibility (or, the degree to which the customer is free to choose the timing of the service) is one of the key prerequisites for successful demand management. At first sight, this condition seems unmet in settings such as an ED. Yet, a certain level of customer flexibility exists, as EDs commonly face non-urgent patients that could also be helped by a general practitioner (so-called inappropriate patients; [41, 45]).

Demand management strategies can serve to complement the capacity strategy. For example, hotels may prefer to employ full-time —and hence more experienced— staff instead of working with temporary employees [199]. Promotions can then be used to ensure sufficient demand throughout the calmer periods such that a level-capacity strategy can be pursued (for personnel capacity). Table 1.2 also contains some options that can be used for *both* capacity management and demand management. Increasing customer participation implies that the service time that requires direct interaction with an employee is reduced (i.e., capacity management), by shifting a part of the workload to the customer (i.e., demand management). A typical example is self-service in fastfood restaurants [79].

The focus on either demand or capacity management depends on the context: more emphasis tends to be placed on demand management when capacity cannot easily be altered (or if it is costly to do so). In the airline industry, for instance, the sheer cost of aircraft has motivated managers to develop strategies that maximize revenue, given a limited amount of seat capacity in an airplane. This approach is known as yield management [146, 48]; it comprises a framework that combines different demand management strategies such as overbooking and price differentiation for different groups of customers, in view of maximizing revenue [146]. For example, price differentiation is often used to discourage customers to arrive during peak periods, or to spur demand in calm periods [212, 119]. Dacko [58] illus-

trate the interdisciplinary nature of demand management and discuss how time-of-day based marketing can contribute to the competitive advantage in service systems with cyclical demand (e.g., coffee shops or supermarkets may offer a different service or assortment based on the time of day).

1.5 Scope and outline of the thesis

In this dissertation, we focus on the staffing and scheduling steps of the capacity planning process (forecasting, rostering, online updating, and demand management fall outside the scope of the dissertation). In particular, we contribute on two aspects of personnel planning with nonstationary arrivals: performance measurement and workforce optimization.

Performance measurement specifies how to evaluate the time-dependent quality of service of a given staffing or scheduling solution. This is highly important when making capacity decisions, because capacity planning models require a performance measurement method as a subroutine, to assess the solution quality of any given plan. Measuring waiting times in systems with nonstationary arrivals is not trivial: the traditional stationary queueing models are likely to be no longer applicable if the arrival rate fluctuates heavily throughout the day. Moreover, the models that evaluate performance in systems with nonstationary arrivals often rely on rather theoretical assumptions and do not capture a number of critical aspects, such as (1) the presence of customer impatience (which causes customers to abandon before receiving service, if their waiting time is too long), (2) the general distribution of service and abandonment times [38], and (3) an exhaustive service policy (where a customer's service is completed, even if this requires the server to work past his scheduled time [121, 51]). Though these features are highly relevant in many practical settings, they are not always included in the analysis (at least, not simultaneously).

Workforce optimization defines how good staffing solutions can be identified in a potentially large solution space. As discussed in Section 1.3, it is common to decompose the workforce optimization process in a staffing and scheduling problem. Current practice in workforce optimization is to solve these problems in a sequential manner [230, 232, 218, 125]. This two-

	Shorter-term DMOs	Longer-term DMOs
Explicit DMOs (options that provide fairly precise control over the arrival of customers)		Scheduling customers Reservations Yield management
Implicit DMOs (options that influence but do not control the arrival of customers)	Price Differentials Offer other incentives during calm periods Service differentials (quality changes depending on the time of day or week) Inform and educate customers about calm periods / alternatives Seek subcontract work	Complementary services Substitute services Advertise to increase business Advertise to achieve a certain demand level Use automation* Provide off-site access (phone-in, internet)* Change the level of customer participation*

*: Can be used for capacity management, demand management, or both

Table 1.2: Demand management options (DMOs)
(source: adapted from [147, 148])

step approach is appealing because the stochastic performance constraint is accounted for during the staffing step; shift scheduling then becomes a deterministic problem. However, as the two-step approach may be far from optimal [120, 110, 111], the focus is recently shifting toward approaches that consider staffing and scheduling as an integrated problem [122, 13].

This dissertation is organized as follows. Chapter 2 offers an extensive literature review on personnel scheduling in systems with nonstationary arrivals. We categorize the available articles based on classification criteria such as the system assumptions, performance evaluation characteristics, optimization approaches and real-life application context. We discuss the main challenges and outline opportunities for continued research. In Chapter 3, we compare several methods to compute time-dependent waiting times in small-scale service systems, with a particular focus on methods that are capable of addressing the following (realistic) problem features: nonstationary arrivals, general service and abandonment distributions, and an exhaustive service policy. Chapters 4 and 5 present solution methods for workforce optimization with nonstationary demand. In Chapter 4, a simulation-based heuristic for personnel staffing (ignoring shift constraints) is presented. Our heuristic succeeds in consistently providing good solutions and does not require strong theoretical assumptions (in contrast to many other heuristics in the literature). Chapter 5 extends the scope by including shift constraints in the analysis: we present a simulation-based branch-and-bound approach for shift scheduling with nonstationary arrivals. The epilogue to this dissertation is presented in Chapter 6. We summarize the main contributions of our research, discuss the insights that have emerged, and elaborate on directions for continued research.

Chapter 2

Staffing and scheduling with nonstationary demand for service: state of the art

2.1 Introduction and scope

In most service systems, staffing drives both costs and service quality. Personnel capacity planning for these systems tends to be non-trivial though, due to the many sources of variability inherent in real-life service systems (e.g., nonstationary demand, stochastic service times, different customer classes) and phenomena like customer abandonment, balking, retrials etc. The personnel capacity planning process usually gets decomposed into four steps [42, 231, 233, 234, 236, 85, 153]:

1. Forecasting demand (based on empirical data).
2. Determining staffing requirements: the staffing levels required over time are selected, in order to meet a specific performance target at minimal cost.
3. Shift scheduling: this step determines how many workers to assign to each shift type, in order to cover the staffing requirements.
4. Rostering: in this final step, employees are assigned to shifts.

Short-term schedule updates may represent an additional step [231, 237, 85] (for an overview and analysis of available methods for online shift updating, see [118, 189, 229]). Demand management, which attempts to manipulate the time-varying arrival rate itself, can be considered as an additional (proactive) process that would precede these capacity planning steps (see [212, 12, 56, 147, 148, 149, 127], among others). Because our goal with this literature review is to provide a state-of-the art overview of research on staffing and personnel scheduling in systems with nonstationary demand, we focus on steps 2 and 3, and consider steps 1 and 4 beyond the scope of this review[1].

The practical relevance of this research field can hardly be overestimated. In many real-life systems (e.g., call centers, emergency departments, toll booths), nonstationary demand is prominent, and appropriate staffing is often the only way to safeguard customer service in these systems. Despite this practical relevance, nonstationary arrivals often do not receive sufficient attention in real-life personnel capacity planning [21, 218, 100]. Consequently, in addition to providing an overview of existing models, their underlying assumptions, and their applicability in practice, we seek to distinguish opportunities for further research that can achieve a better integration of theory and practice.

This research field has grown rapidly in the past two decades. We focus on the period 1991-2013, selecting 61 articles that *focus on personnel staffing and/or scheduling* and that *specifically target systems with nonstationary demand (i.e., stochastic with a time-varying rate).* Table 2.1 gives an overview of the selected articles. We categorize these based on four classification criteria: system assumptions, performance evaluation characteristics, optimization approaches and real-life application context. We did not include in the categorization articles that present general staffing or scheduling algorithms for deterministic and/or non-time-varying systems (e.g., [250, 198, 22, 16, 27, 151]), or that focus solely on scheduling algorithms, with assumptions of exogenous staffing requirements (as in the early

[1] See [85, 209, 186, 9, 7, 241] for issues related to demand forecasting. A more elaborate discussion of the rostering problem can be found in De Causmaecker and Vanden Berghe [62] and Burke et al. [43].

work of Dantzig [60] and Keith [140]). Similarly, we excluded manuscripts that centered on other types of resources (such as hospital beds; [259]).

Time range	Number of articles	References
1991 - 1995	4	[2], [11], [200], [230]
1996 - 2000	10	[82], [110], [111], [115], [131], [157], [164], [182], [232], [248]
2001 - 2005	10	[13], [23], [47], [83], [96], [108], [120], [137], [155], [211]
2006 - 2010	24	[24], [5], [14], [17], [25], [28], [31], [32], [46], [52], [72], [75], [78], [98], [105], [107], [109], [122], [139], [143], [170], [203], [210], [218]
2011 - 2013	13	[44], [55], [65], [68], [86], [125], [144], [161], [162], [168] [193], [208], [258]

Table 2.1: Categorized articles

This overview differs in some key respects from previously published review articles in this field. For example, Gans et al. [85] and Aksin et al. [7] present surveys that specifically target call centers, discussing not only staffing problems but also various other operational problems related to this specific application area. Our review focuses solely on staffing and scheduling for nonstationary demand systems, and we discuss the relevance of different models to various application areas. Green et al. [99] and Whitt [253] offer an extensive overview of methods for staffing with nonstationary demand for service, but the methods they propose rely largely on stationary approximations (see Section 2.4.1) and do not include shift scheduling. Ernst et al. [73, 74] and more recently Van den Bergh et al. [240] provide comprehensive reviews of research on scheduling and rostering, but do not specifically focus on methods for nonstationary demand. We consider both staffing and scheduling, in settings with nonstationary stochastic demand for service.

The remainder of this chapter is organized as follows: Section 2.2 describes the classification scheme used to categorize the literature. The following sections then provide an in-depth discussion of each classification

criterion. Section 2.3 features the classification of the articles in accordance with the system assumptions, and Section 2.4 outlines the evaluation methods for system performance. Because performance evaluation is necessary to evaluate proposed solutions and guide the search for better solutions, it is a highly relevant subroutine in any staffing or shift scheduling approach. We offer an overview of the optimization methodologies in Section 2.5, then classify the articles on the basis of the suggested real-life application areas in Section 2.6. Finally, Section 2.7 contains the conclusions and identifies promising directions for further research.

2.2 Overview of classification criteria

Figure 2.1 displays a simple representation of a (single-stage) service system with nonstationary demand. Customers arrive according to a nonstationary arrival process with time-varying arrival rate λ_t (where t represents time). Typically, the arrival pattern repeats over a given cycle (e.g., day, week, month, year). The service process starts immediately if a server is available on arrival; otherwise, the customer joins the queue. The aggregate service rate (denoted $s_t\mu$) can be influenced by changing s_t, the number of servers available at time t. The per server service rate μ is commonly assumed to be constant, though some models allow for time-varying service rates (e.g., [131]).

In many service systems, customers may opt to abandon by leaving the queue without being served; they are referred to as *abandonments* (or *left without being seen*, or LWBS, in a healthcare context). Long waiting times are the main reason for customers to abandon (Johnson et al. [134] report that almost 77% of LWBS patients in an emergency department claim to abandon because of long waiting times). Although abandonments are undesirable from a customer service perspective, they tend to have a positive effect on system stability, especially when the system is temporarily overloaded (e.g., [99]). The abandoned customers are also not, by definition, "lost", because unserviced customers may reenter the system later. *Retrials* refer to customers that abandoned previously either upon arrival (because the queue was too long [3, 4]), or after experiencing a positive waiting time

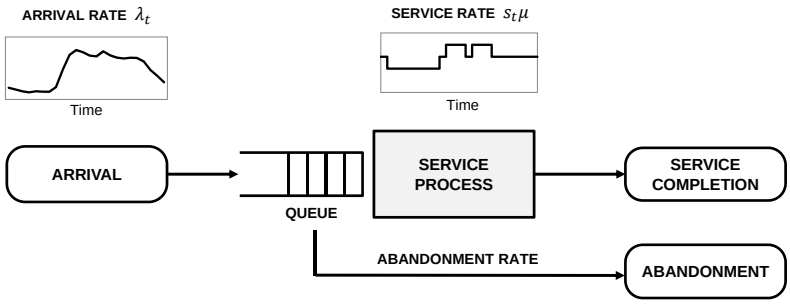


Figure 2.1: Schematic representation a single-stage queueing system with nonstationary demand.

[180]. If there are no retrials, ignoring abandonment behavior tends to cause overstaffing, implying higher labor costs. Note that serviced customers may also reenter the queue if they need to be serviced several times by the same server (*reentrant customers*, see [255, 154]).

It is possible to classify previous publications by the criteria listed in Table 2.2: system assumptions, performance evaluation characteristics, optimization approaches and real-life application context.

For the system assumptions classifier, we rely on the commonly used Kendall notation [142] to reflect any assumptions regarding the arrival and service processes in the system. Heyman and Whitt [113] were among the first to add the notation "t" to represent the time-dependent nature of the arrival process; the notation for customer abandonments was introduced by Baccelli and Hebuterne [20]. For example, the $M_t/G/s_t + G$ notation represents a system with time-varying Poisson arrivals (M_t), a general service time distribution (the first G), time-varying staffing levels s_t, and abandonments that follow a general distribution (the last G). Other relevant features are the homogeneity of customers and/or servers, the presence of staffing intervals, the queueing discipline, the service discipline, the structure of the system, and parameter uncertainty. Customers are heterogenous if the system takes different customer classes into account (e.g., due to differences in process steps, service times, or queueing discipline [84]); if only

Classifier	Features	Notation
System assumptions	Kendall notation	A/B/C+L with A = distribution of the arrival process, B = distribution of service process, C = number of servers, L = distribution of the abandonment process.
	Homogeneity of customers / servers	HO = homogenous and HE = heterogenous.
	Staffing interval	Y = yes; N = no.
	Queueing policy	FIFO = first-in first-out; SBR = skill-based routing; Priority = queueing based on customer priority.
	Service policy	E = exhaustive; P = preemptive.
	System structure	S = single stage; N = network.
	Parameter uncertainty	Y = yes; N = no.
Performance evaluation	Methodology	
	Performance metrics	see Table 2.4
Optimization approach	Methodology	
	Objective	
	Constraints	
Real-life application context	Context	
	Implementation (+results reported)	Y=yes; N=no.
	Validation by means of real-life data	Y=yes; N=no.
	Validation by means of other (fictive) examples	Y=yes; N=no.

Table 2.2: Overview of classifiers, features and notation

a single customer class is considered, customers are homogenous. Servers are homogenous if they all exhibit the same skills (i.e., can all handle the same types of customers at the same rate) and have the same service rate; otherwise, they are heterogenous servers.

A common assumption is that capacity changes can be made only at specific points in time; the time period during which capacity remains constant is the *staffing interval*. The staffing interval length can vary: e.g., Defraeye and Van Nieuwenhuyse [65] use an interval length of 15 minutes in their computational results, whereas Izady and Worthington [125] use intervals of 30 minutes or 1 hour (the methods can equally be applied to other staffing interval lengths).

The queueing policy refers to the sequence in which customers are serviced; first-in first-out (FIFO) is by far the most frequently used queueing discipline in the articles we survey, though priority-based rules are also common, particularly in the context of emergency services (e.g., priority based on the urgency of a patient's condition). The service policy reflects what happens to a customer in service when a server is scheduled to leave. Many existing models implicitly assume a preemptive service discipline [123], such that service is interrupted and the customer in service rejoins the queue. Under the (more realistic) exhaustive service policy, the customer service instead gets completed before the server leaves, even if this means that a server has to work beyond his or her scheduled time.

For system structure, we distinguish between systems that contain only a single service step (single-stage models) and those that contain multiple service steps (networks). Finally, we check whether the model accounts for parameter uncertainty. The use of stochastic arrival rates, service rates, and abandonment rates requires an estimation of the distributional parameters, which might introduce error into the models (and cause the desired performance target to be violated). Accounting for this parameter uncertainty during the personnel capacity planning process can significantly improve the staffing solutions (though possibly at a higher staffing cost; [203, 161]).

For the performance evaluation classifier, we categorize prior contributions according to the methodology used to evaluate the performance of a given personnel allocation, that is, given s_t values. We provide key ref-

erences for each evaluation method. In addition, we list the performance metrics and discuss which metrics are most common in practice, in different application contexts.

By considering the optimization approach, we can categorize contributions according to the methodology used to optimize personnel capacity, along with the objective and the constraints. Models that vary s_t without taking into account shift requirements (e.g., shift patterns, shift durations) are staffing models (they result in *staffing requirements*); otherwise, they are shift scheduling models. A common approach to shift scheduling is to first determine the staffing requirements necessary to meet the desired performance at minimum cost, then fit the minimum cost shift schedule to these staffing requirements (either by considering the staffing requirements as a strict lower bound on the capacity level at each moment in time, or by interpreting them as a guideline, such that the capacity levels defined by the shift schedule should adhere closely to the staffing requirements). We refer to this method as the *two-step approach*. However, the two-step approach may lead to suboptimal shift schedules [120, 110, 111], because several equivalent staffing solutions might exist that lead to shift schedules with substantially varying costs [122, 230, 13]. Therefore, the recent literature increasingly focuses on methods that either address the staffing and scheduling problem simultaneously (i.e., the *integrated approach*), or that skip the staffing step by scheduling shifts directly according to the nonstationary demand (we call this the *direct approach*). A detailed description of these approaches is provided in Section 2.5.2.

Finally, for the real-life application category, we classify articles on the basis of their application context, as suggested by the authors, as well as according to evidence of real-life implementation, validation using real-life data, or validation using other (fictive) examples.

2.3 Classification by system assumptions

Table 2.3 displays the literature classification based on the system assumptions. These assumptions are often linked with the choice of a performance evaluation method and/or capacity optimization approach, as discussed fur-

ther in Sections 2.4 and 2.5, respectively.

A large majority of extant studies assume that both customer types and server types are homogenous and that the system consists of a single stage. More recent work has shifted this emphasis toward models that include both customer and server heterogeneity (albeit with exponential assumptions on the service and abandonment time distribution, see Table 2.3), as is further detailed below. The few articles that consider a service network, assume that customers and servers are heterogenous; none of these studies include abandonments.

Homogeneity customers / servers	Kendall notation	References	Staffing interval (Y/N)	Queueing policy	Service policy (E/P)	System structure (S/N)	Parameter uncertainty (Y/N)
HO/HO	$M_t/M/s_t$	[2]	Y	FIFO*	(n.s.)	S	N
		[55]	Y	FIFO*	(n.s.)	S	N
		[72]	Y	(n.s.)	(n.s.)	S	N
		[96]	Y	FIFO*	(n.s.)	S	N
		[98]	Y	FIFO*	P	S	N
		[120]	Y	FIFO*	(n.s.)	S	N
		[122]	Y	FIFO*	P	S	N
		[143]	Y	FIFO*	(n.s.)	S	N
		[155]	Y	FIFO*	(n.s.)	S	N
		[162]	Y	FIFO	(n.s.)	S	Y
		[208]	Y	FIFO	P/E	S	Y
		[230]	Y	FIFO*	E	S	N
		[232]	Y	FIFO	(n.s.)	S	N
	$M_t/M/s_t + M$	[68]	Y	FIFO*	(n.s.)	S	N
		[86]	Y	FIFO*	(n.s.)	S	Y
		[107]	N (+Y)	FIFO*	(n.s.)	S	N
		[111]	Y	FIFO*	(n.s.)	S	N
		[144]	Y	FIFO	P	S	N
		[203]	Y	FIFO*	(n.s.)	S	Y
		[211]	Y	FIFO*	(n.s.)	S	N
		[210]	Y	FIFO*	(n.s.)	S	N
	$M_t/M/s_t + G$	[11]	Y	FIFO*	(n.s.)	S	N
		[200]	Y	FIFO*	(n.s.)	S	N
	$M_t/G/s_t$	[13]	Y	FIFO	E	S	N
		[14]	Y	FIFO	E	S	N
		[248]	N	FIFO	(n.s.)	S	Y
	$M_t/G/s_t + G$	[65]	Y	FIFO	E	S	N
		[78]	N	FIFO*	(n.s.)	S	N
		[170]	N (+Y)	FIFO	P (+ E)	S	N
	$G_t/M/s_t$	[82]	Y	FIFO*	(n.s.)	S	N
	$G_t/G_t/s_t$	[131]	Y	FIFO*	(n.s.)	S	N
	$G_t/G/s_t + G$	[44]	Y	FIFO*	P/E	S	N
		[168]	N	FIFO	(n.s.)	S	N
	Not specified	[46]	Y	FIFO*	(n.s.)	S	N
		[52]	Y	FIFO*	(n.s.)	S	N
		[110]	Y	FIFO*	(n.s.)	S	N

(continued on next page)

(continued from previous page)

Homogeneity customers / servers	Kendall notation	References	Staffing interval (Y/N)	Queueing policy	Service policy (E/P)	System structure (S/N)	Parameter uncertainty (Y/N)
		[139]	Y	FIFO*	(n.s.)	S	N
		[157]	Y	FIFO*	(n.s.)	S	N
		[182]	Y	FIFO*	(n.s.)	S	N
		[193]	Y	FIFO*	(n.s.)	S	N
HO/HE	$M_t/M/s$	[31]	-	FIFO*	(n.s.)	S	Y
	$M_t/M/s_t + M$	[115]	Y	FIFO	(n.s.)	S	N
	$M_t/G/s_t + G$	[164]	Y	FIFO*	(n.s.)	S	N
	$M_t/M/s_t + M$	[137]	Y	FIFO*	(n.s.)	S	Y
HE/HE	$M_t/M/s_t$	[17]	Y	FIFO / priority	(n.s.)	S	N
		[32]	Y	SBR	(n.s.)	S	N
		[75]	Y	FIFO*	(n.s.)	S	N
		[161]	N	FIFO	(n.s.)	S	Y
		[258]	Y	Priority	(n.s.)	N	N
	$M_t/M/s + M$	[24]	-	SBR	E	S	Y
		[25]	-	SBR	P	S	Y
		[108]	-	SBR	P	S	Y
	$M_t/M/s_t + M$	[23]	Y	SBR	P	S	Y
		[28]	Y	SBR	(n.s.)	S	Y
		[105]	Y	SBR	(n.s.)	S	Y
		[109]	Y	SBR	P	S	Y
	$M_t/G/s$	[5]	-	Priority*	(n.s.)	N	N
	$M_t/G/s_t$	[47]	Y	FIFO*	(n.s.)	N	N
		[125]	Y	Priority	E	N	N
	Not specified	[83]	Y	SBR*	(n.s.)	N	N
		[218]	Y	Priority*	(n.s.)	N	N

Table 2.3: Classification by system assumptions

It is worthwhile to explore in further detail the classification according to Kendall notation, irrespective of the other assumptions, as in Figure 2.2. It shows that the large majority of contributions have focused on systems with time-varying number of servers. Among these, the $M_t/M/s_t$ model can be considered as a "base" model, which can then be extended by including abandonments, and/or changing exponential distribution assumptions into general distributions. The figure highlights that the inclusion of Poisson abandonments (yielding the $M_t/M/s_t+M$ model) has received considerable attention, while the extension towards general distributions is less common (because performance evaluation then becomes more complex). An overwhelming majority of articles assumes a nonstationary Poisson arrival process; Kim and Whitt [145] find that this assumption is consistent with empirical arrival processes observed in call centers and emergency departments. Daily recurring demand patters typically display one to three peaks

per day [230, 97]. Authors frequently resort to sine functions to generate demand rate profiles for their computational experiments: see for example Green et al. [96], Liu and Whitt [170] (only one peak per cycle) and Ingolfsson et al. [122], Green et al. [97] (two peaks per cycle). The applicability of the staffing and scheduling models, however, does not depend on the use of the sine function. Many methods actually assume that the arrival rate is constant over the staffing interval [96], and therefore average the arrival rate over that interval (a more restrictive approach instead considers the maximum arrival rate over the staffing interval, [96]). This is reasonable because real-life data are often available only on an aggregate basis, e.g., per hour or half hour [125, 98, 5, 258].

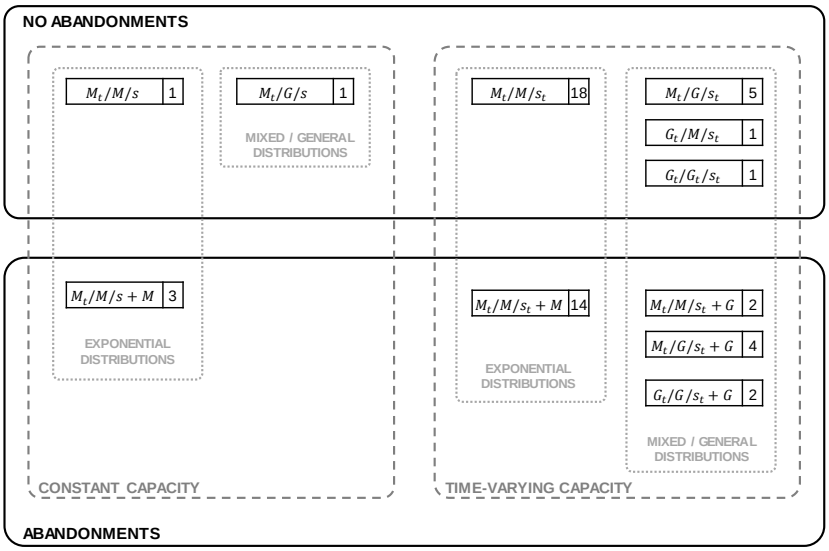


Figure 2.2: Classification based on Kendall notation (number of articles).

Figure 2.2 also reveals that a majority of published articles assume the service process is exponentially distributed. Zeltyn et al. [258] and Hueter and Swart [115] largely validate this assumption using empirical data for an emergency department and restaurant setting, while non-exponential service time distributions have been reported in a call center context (e.g., Brown et al. [38] report a lognormal distribution and Castillo et al. [46] report

Erlang distributed service times). Abandonments, if included at all, are also commonly assumed to follow an exponential distribution. It is known that, in systems with abandonments, the impact of the exact choice of the service and abandonment distributions depends on the system utilization. In *stationary* systems the service time distribution is more important than the patience time distribution when the systems are critically loaded [59, 175], and the patience time distribution is more important than the service time distribution when the systems are overloaded [249, 251]. Recently however, Chassioti et al. [50] suggest that, in systems with *nonstationary* demand and abandonment, the distribution of service time beyond its mean is relatively unimportant.

Table 2.3 shows that the queueing policy is predominantly FIFO; only when both customers and servers are heterogenous do we find evidence of priorities or skill-based routing (SBR). In practice, the use of priorities is common particularly in health care settings [125, 258], whereas call center models mostly rely on skill-based routing, which impacts the sequence in which customers receive service[2]. Accounting for customer routing adds complexity to the personnel capacity decision process, in the sense that the system's performance depends on not only staffing (or scheduling) decisions, but also routing decisions. Harrison and Zeevi [108], Bassamboo and Zeevi [24, 25], and Bertsimas and Doan [28], among others, propose methods to solve the staffing and (dynamic) routing problems in call centers with heterogeneous servers and customers. Bassamboo and Zeevi [23] extend their previous work [24] by including admission control decisions.

Many articles fail to provide details on the service policy being applied. According to Ingolfsson et al. [121], extensive literature (implicitly) assumes a preemptive service discipline, whereas in many real-life settings, the service policy is inherently exhaustive [121, 51, 77, 123, 44]. The service policy is likely to have a large impact if the average service time is relatively long compared to the staffing interval: the amount of overtime evidently depends on the service time of the customer being in service, while shorter staffing intervals imply that capacity decreases are more frequent (these potentially

[2] An overview of problems related to staffing and routing in call centers can be found in [153].

initiate overtime). The effect of the service policy will be less prominent in systems with low average utilization though, because servers are then more likely to be idle at the end of their shift.

Table 2.3 also shows that most articles do not account for parameter uncertainty[3] or network settings. As shown by Table 2.4 (that details those system assumptions for which we observed an evolution over time) these two phenomena have only appeared very recently in the literature. The move towards heterogenous customer and server settings (HE/HE) can be seen as another recent trend. Accounting for parameter uncertainty during the personnel capacity planning process can lead to significant reductions in the total expected cost (which generally includes, besides the personnel cost, a penalty for not meeting the performance constraint; [203, 161]). As is evident from Table 2.3 though, staffing and scheduling models that include parameter uncertainty tend to rely on exponential assumptions for the arrival, service, and abandonment processes.

		1991 - 1995	1996 - 2000	2001 - 2005	2006 - 2010	2011 - 2013
Homogeneity customers / servers	HO/HO	4	8	5	13	10
	HO/HE	0	2	1	1	0
	HE/HE	0	0	4	10	3
Service policy	Preemptive	0	0	2	5	3
	Exhaustive	1	0	1	3	4
	Not specified	3	10	7	17	8
System structure	Single stage	4	10	8	22	11
	Network	0	0	2	2	2
Parameter uncertainty included	Yes	0	1	3	7	4
	No	4	9	7	17	9

Table 2.4: Trends in system assumptions (number of articles)

A final observation from Table 2.3 is that, while a considerable number of models models include some type of additional complexity (e.g., by considering non-exponential service and abandonment times, non-homogenous

[3]For general references on the impact and implications of parameter uncertainty, see [51, 173, 204, 220, 221, 205].

customers or servers, network settings, etc.), we found no articles that address all aspects simultaneously. Moreover, we observe that extensions toward networks of queues and exhaustive service policies are particularly underrepresented in the literature, and present challenging directions for future research.

2.4 Classification by performance evaluation methods and performance metrics

This section highlights the performance metrics evaluated in each article, and classifies articles according to the methodology used to evaluate system performance for given capacities.

The number of performance metrics actually used is vast, as the overview in Table 2.4 reveals (this table also clarifies the more concise notation we use in Tables 2.6, 2.7 and 2.8). We distinguish metrics based on number in system/number in queue, waiting time, abandonments/throughput, length of stay, and utilization[4]. In terms of notation, we closely adhere to that introduced in Baron and Milner [22]: we distinguish between metrics taken over the planning horizon (horizon-based, $(\cdot)_{\mathrm{HB}}$), those assessed over a smaller interval such as a staffing interval (interval-based, $(\cdot)_{\mathrm{IB}}$), and instantaneous metrics (time epoch-based, $(\cdot)_{\mathrm{TB}}$). Metrics that are based on per customer performance are represented as $(\cdot)_{\mathrm{CB}}$ (customer-based).

Notation	Interpretation
Number in system / queue:	
N_t	Number in system at time t
B_t	Number busy servers at time t
Q_t	Queue length at time t
$P_{\mathrm{TB}}(Q \geq q)$	Queue length tail probability
$E_{\mathrm{TB}}[Q]$	Expected number in queue, at time t
$E_{\mathrm{IB}}[Q]$	Expected queue length, over a given staffing interval
$E_{\mathrm{HB}}[Q]$	Expected queue length, over time horizon T
$\max_{\mathrm{HB}}\{Q\}$	Maximum queue length measured over time horizon T

(continued on next page)

[4]We do not explicitly include labor cost as a performance metric, because its calculation is usually straightforward.

(continued from previous page)

Notation	Interpretation
$E_{\mathrm{HB}}[E_{\mathrm{IB}}[Q]]$	Expected queue length, measured over a given staffing interval and averaged over time horizon T
$E_{\mathrm{HB}}[N]$	Expected number in system (in queue and in service) over time horizon T
Waiting time:	
$P_{\mathrm{TB}}(W > 0)$	Probability of experiencing a positive waiting time, upon arrival at time t
$P_{\mathrm{IB}}(W > 0)$	Probability of experiencing a positive waiting time, upon arrival in a given staffing interval
$E_{\mathrm{TB}}[W]$	Expected waiting time, at time t
$E_{\mathrm{IB}}[W]$	Expected wait, measured over a given staffing interval
$E_{\mathrm{HB}}[W]$	Expected waiting time, over time horizon T
$\max_{\mathrm{HB}}\{W\}$	Maximum wait, measured over time horizon T
$E_{\mathrm{HB}}[C_{\mathrm{CB}}(W > 0)]$	Expected cost for positive wait
$E_{\mathrm{HB}}[C_{\mathrm{CB}}(W)]$	Expected cost for length of waiting time
$P_{\mathrm{TB}}(W > \tau)$	Probability of experiencing a waiting time exceeding τ, upon arrival at time t
$P_{\mathrm{IB}}(W > \tau)$	Probability of experiencing a waiting time exceeding τ, upon arrival in a given staffing interval
$P_{\mathrm{HB}}(W > \tau)$	Probability of experiencing a waiting time exceeding τ, for all arrivals over time horizon T
$E_{\mathrm{HB}}[P_{\mathrm{TB}}(W > \tau)]$	Probability of experiencing a waiting time exceeding τ, upon arrival at time t, averaged over time horizon T
$E_{\mathrm{HB}}[P_{\mathrm{IB}}(W > \tau)]$	Probability of experiencing a waiting time exceeding τ, upon arrival in a given staffing interval , averaged over time horizon T
$E_{\mathrm{TB}}[W - \tau \mid W > \tau]$	Average excess with regard to maximum allowed waiting time τ
$\frac{\int_0^T \lambda_t P_{\mathrm{TB}}(W \leq \tau) dt}{\int_0^T \lambda_t dt}$	Weighted service level
$\min_{\mathrm{HB}}\{P_{\mathrm{TB}}(W > \tau)\}$	Minimal service level over time horizon T
$E_{\mathrm{IB}}[CGOS]$	Expected customer grade of service per interval (utility function based on waiting time)
$E_{\mathrm{HB}}[\mathrm{Coverage}]$	Expected aggregated coverage, over time horizon T
$E_{\mathrm{IB}}[\mathrm{Coverage}]$	Expected coverage, over a given staffing interval
Abandonments / throughput:	
Ab_t	Abandonment rate, as a function of t
$P_{\mathrm{TB}}(Ab)$	Abandonment probability, as a function of t
$E_{\mathrm{HB}}[\%Ab]$	Average percentage abandoned, over time horizon T
$E_{\mathrm{IB}}[\%Ab]$	Expected percentage abandoned, over a given staffing interval
$E_{\mathrm{HB}}[E_{\mathrm{IB}}[\%Ab]]$	Expected percentage abandoned, measured over a given staffing interval and averaged over time horizon T
Bl_t	Blocking rate, as a function of t
$E_{\mathrm{HB}}[\%Bl]$	Expected percentage blocked, over time horizon T
$E_{\mathrm{HB}}[\%\mathrm{Served}]$	Fraction of customers that is served, over time horizon T

(continued on next page)

(continued from previous page)

Notation	Interpretation
$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$	Expected abandonment cost, over time horizon T
$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Bl)]$	Expected blocking cost, over time horizon T
$\mathrm{throughput}_t$	Throughput, as a function of t
$E_{\mathrm{IB}}[\mathrm{throughput}]$	Expected throughput over a given staffing interval
$E_{\mathrm{HB}}[\mathrm{throughput}]$	Expected throughput over time horizon T
Length of stay:	
$E_{\mathrm{HB}}[LoS]$	Expected length of stay, over time horizon T
$P_{\mathrm{HB}}(LoS < \alpha)$	Probability of experiencing a length of stay exceeding α, over time horizon T
Utilization	
U_t	Utilization, as a function of t
$E_{\mathrm{IB}}[U]$	Expected utilization over a given staffing interval
$E_{\mathrm{HB}}[U]$	Expected utilization, over time horizon T
$E_{\mathrm{HB}}[E_{\mathrm{IB}}[U]]$	Expected utilization, measured over a given staffing interval and averaged over time horizon T
SIT_{HB}	Server idle time, over time horizon T
$E_{\mathrm{TB}}[Busy]$	Expected number of busy servers at time t
Number of hours where $U_{\mathrm{TB}} > u$	Number of hours utilization exceeds a certain percentage, over time horizon T
$\max_{\mathrm{HB}}[U]$	Maximum utilization, over time horizon T

Table 2.5: Overview of performance metrics and compact notation

Table 2.6 contains the performance evaluation metrics and methodologies for the studied articles; it highlights that the performance metrics tend to depend on the application context. Often, specific terminology then applies. In emergency departments, waiting times and length-of-stay (LoS) metrics are most common. Abandonments are commonly referred to as *left without being seen* or LWBS [98]. Call centers tend to focus either on the service level (which is then referred to as the *total service factor* or TSF, [203]) or the expected waiting time (*average speed of answer* or ASA, e.g., [211]). The category "other" in Table 2.6 includes references on personnel scheduling in restaurants [115, 52], crew scheduling for ambulances [72], personnel scheduling in retail stores [139, 157], and scheduling customs staff at airports [182]. Many metrics in these contexts relate to service levels: in ambulance scheduling, the *coverage* (which specifies the probability that the response time lies below a given time limit) is maximized [72]. In retail,

on the contrary, a profit-driven approach is common. For instance, Lam et al. [157] consider profit as sales revenue minus personnel cost, and model sales revenue as a function of personnel staffing, customer arrivals, and other factors. Customer service is checked afterwards, by measuring the *service availability* for the resulting schedule (expressed by the ratio of staff number to traffic). In restaurants, Hueter and Swart [115] aim to limit the expected waiting time and percentage abandoned customers, whereas Choi et al. [52] target a constant ratio of customers to servers (*customer count per server*, or CCS).

Some authors seek to exploit the relation between performance metrics, using performance metrics that are easy to compute to obtain results for more complex performance metrics. Simply-computed metrics are often sufficient to guide the search for adequate personnel schedules, e.g., Izady and Worthington [125] apply analytic results related to delay probability to determine shift schedules that meet a length-of-stay target in an emergency department. Similarly, Green et al. [98] focus on a service level (at most 20% of patients wait more than 1 hour) to realize a reduction in the percentage LWBS. Kim and Ha [144] impose an upper bound on the number of customers in the call center, which is used as a proxy metric to control the expected waiting time, the delay probability and the service level. Exploring the relationships across different performance metrics in complex nonstationary systems may open up interesting opportunities for further research, particularly for performance metrics that are difficult to compute.

We elaborate on the performance evaluation methodologies in the following sections. Section 2.4.1 describes how stationary models can be applied to estimate performance in systems with nonstationary arrivals. Section 2.4.2 discusses discrete-event simulation and 2.4.3 addresses numerical methods (such as randomization and discrete-time modeling). Fluid approximations are described in Section 2.4.4. Section 2.4.5 briefly elaborates on how empirical data have been used for performance evaluation.

Context	References	Stationary approximation	Discrete event simulation	Numerical methods	Fluid model	Empirical	Performance metrics related to ...				
							Number in system / queue	Waiting time	Abandonment / throughput	LoS	Utilization
General	[44]	IS						$P_{\mathrm{TB}}(W > 0)$, $E_{\mathrm{TB}}[W]$, $P_{\mathrm{TB}}(W \leq \tau)$, $E_{\mathrm{TB}}[W-\tau \mid W > \tau]$, $\min_{\mathrm{HB}}\{P_{\mathrm{TB}}(W > \tau)\}$, $E_{\mathrm{HB}}[P_{\mathrm{TB}}(W > \tau)]$	$P(Ab_t)$		
	[46]		x				$E_{\mathrm{HB}}[Q]$, $\max_{\mathrm{HB}}\{Q\}$	$E_{\mathrm{HB}}[W]$, $\max_{\mathrm{HB}}\{W\}$, $P_{\mathrm{HB}}(W > \tau)$	$E_{\mathrm{HB}}[\%Ab]$, $E_{\mathrm{HB}}[\%Bl]$		$E_{\mathrm{HB}}[U]$
	[78]		x					$P_{\mathrm{TB}}(W > 0)$			
	[82]			x			$E_{\mathrm{HB}}[N]$				
	[96]	lag SIPP						$P_{\mathrm{IB}}(W > 0)$			
	[120]			x				$P_{\mathrm{TB}}(W > 0)$, $P_{\mathrm{TB}}(W > \tau)$			
	[122]			x				$P_{\mathrm{TB}}(W \leq \tau)$, $\min_{\mathrm{HB}}\{P_{\mathrm{TB}}(W > \tau)\}$			
	[131]	IS					$E_{\mathrm{TB}}[Q]$	$P_{\mathrm{TB}}(W > 0)$			
	[170]	MOL/IS			x		$E_{\mathrm{TB}}[Q]$	$E_{\mathrm{TB}}[W]$, $P_{\mathrm{TB}}(W > 0)$	$P_{\mathrm{TB}}(Ab)$		$E_{\mathrm{TB}}[Busy]$
	[168]				x		Q_t, N_t, B_t	$E_{\mathrm{TB}}[W]$			
	[230]	EAR						$E_{\mathrm{HB}}[P_{\mathrm{IB}}(W \leq \tau)]$			
	[232]	EAR						$P_{\mathrm{HB}}(W < \tau)$, $P_{\mathrm{IB}}(W < \tau)$			
Emergency department- ment	[5]		x					$E_{\mathrm{HB}}[W]$	E_{HB}[throughput]		
	[47]		x							$E_{\mathrm{HB}}[LoS]$	
	[65]		x					$P_{\mathrm{TB}}(W > \tau)$, $P_{\mathrm{TB}}(W > 0)$, $E_{\mathrm{IB}}[W]$			
	[98]	lag SIPP						$P_{\mathrm{IB}}(W > \tau)$	$E_{\mathrm{HB}}[\%Ab]$		
	[125]	MOL	x					$P_{\mathrm{TB}}(W > 0)$, $E_{\mathrm{IB}}[W]$		$P_{\mathrm{HB}}(LoS < 4h)$	U_t

(continued on next page)

(continued from previous page)

Context	References	Stationary approximation	Discrete event simulation	Numerical methods	Fluid model	Empirical	Performance metrics related to ... Number in system / queue	Waiting time	Abandonment / throughput	LoS	Utilization
	[218]		x					$P_{\mathrm{HB}}(W > \tau)$, $P_{\mathrm{IB}}(W > \tau)$, $E_{\mathrm{IB}}[W > \tau)]$		$E_{\mathrm{HB}}[LoS]$	U_t, Number of hours where $U_{\mathrm{TB}} > u$, $\max_{\mathrm{HB}}[U]$
	[258]	MOL	x							$E_{\mathrm{HB}}[LoS]$	$E_{\mathrm{IB}}[U]$
Call center	[24]	PSA			x			$E_{\mathrm{HB}}[C_{\mathrm{CB}}(W)]$	$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$		
	[2]	SIPP*				x		$P_{\mathrm{HB}}(W < \tau)$	$P_{\mathrm{HB}}(\%Ab)$,		
	[11]	SIPP*						$P_{\mathrm{HB}}(W < \tau)$	$P_{\mathrm{HB}}(\%Bl)$		
	[13], [14]		x					$P_{\mathrm{IB}}(W > 0)$, $P_{\mathrm{IB}}(W > \tau)$,			
	[17]		x					$P_{\mathrm{IB}}(W > \tau)$, $P_{\mathrm{HB}}(W > \tau)$			
	[23]	PSA			x			$E_{\mathrm{HB}}[C_{\mathrm{CB}}(W)]$	$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Bl)]$,		
	[25]	PSA			x			$E_{\mathrm{HB}}[C_{\mathrm{CB}}(W)]$	$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$		
	[28]				x			$E_{\mathrm{HB}}[C_{\mathrm{CB}}(W)]$,	$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$		
	[31]			x				$E_{\mathrm{HB}}[C_{\mathrm{CB}}(W > 0)]$			
	[32]	SIPP*					$E_{\mathrm{IB}}[Q]$,	$P_{\mathrm{IB}}(W < \tau)$	$E_{\mathrm{IB}}[\%Ab]$,		$E_{\mathrm{IB}}[U]$,
	[55]		x					$P_{\mathrm{IB}}(W > 0)$			
	[68]	SIPP					$E_{\mathrm{HB}}[E_{\mathrm{IB}}[Q]]$	$P_{\mathrm{IB}}(W > \tau)$, $E_{\mathrm{HB}}[P_{\mathrm{IB}}(W > \tau)]$	$E_{\mathrm{HB}}[E_{\mathrm{IB}}[\%Ab]]$		$E_{\mathrm{HB}}[E_{\mathrm{IB}}[U]]$
	[75]	SIPP	x					$P_{\mathrm{IB}}(W \leq \tau)$			
	[83]	SIPP*	x					$P_{\mathrm{HB}}(W > \tau)$	$E_{\mathrm{HB}}[\%Ab]$		SIT_{HB}
	[86]	SIPP*							$E_{\mathrm{IB}}[\%Ab]$,		
									$E_{\mathrm{HB}}[\%Ab]$		
	[105]		x						$P_{\mathrm{HB}}(Ab_t > \alpha)$		
	[107]	MOL			x				$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$,		
									$E_{\mathrm{HB}}[\%Ab]$,		
									$E_{\mathrm{HB}}[\%Bl]$,		
									$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Bl)]$		

(continued on next page)

(continued from previous page)

Context	References	Stationary approximation	Discrete event simulation	Numerical methods	Fluid model	Empirical	Performance metrics related to ...				
							Number in system / queue	Waiting time	Abandonment / throughput	LoS	Utilization
	[108]				x				$E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$, throughput$_t$		
	[109]		x		x			$E_{\mathrm{HB}}[W]$	E_{HB}[%Served]		
	[110]		x					$E_{\mathrm{IB}}[CGOS]$			
	[111]	PSA						$E_{\mathrm{IB}}[CGOS]$			
	[137]	SIPP						$P_{\mathrm{IB}}(W \leq \tau)$			
	[143]		x					$P_{\mathrm{IB}}(W \leq \tau)$			
	[144]	IS					Q_t				
	[155]	SIPP						$P_{\mathrm{IB}}(W > \tau)$, $E_{\mathrm{HB}}[P_{\mathrm{IB}}(W > \tau)]$			
	[161]	SIPP						$P_{\mathrm{TB}}(W \leq \tau)$			
	[162]	SIPP						$P_{\mathrm{HB}}(W \leq \tau)$			
	[164]	SIPP	x			x			$E_{\mathrm{IB}}[\%Ab]$		
	[193]					x		$E_{\mathrm{IB}}[W]$, $E_{\mathrm{HB}}[W]$	$E_{\mathrm{IB}}[\%Ab]$, $E_{\mathrm{HB}}[\%Ab]$		
	[200]	SIPP*				x		$E_{\mathrm{HB}}[W]$, $P_{\mathrm{HB}}(W < \tau)$	$P_{\mathrm{HB}}(\%Ab)$, $P_{\mathrm{HB}}(\%Bl)$		
	[203]	SIPP*						$P_{\mathrm{HB}}(W \leq \tau)$			
	[208]		x	x				$E_{\mathrm{HB}}[P_{\mathrm{TB}}(W \leq \tau)]$, $P_{\mathrm{IB}}(W \leq \tau)$			
	[211]	SIPP	x				$E_{\mathrm{HB}}[Q]$	$P_{\mathrm{IB}}(W > 0)$, $E_{\mathrm{HB}}[C_{\mathrm{CB}}(W > 0)]$	$E_{\mathrm{HB}}[\%Ab]$, $E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$		
	[210]	SIPP	x				$E_{\mathrm{HB}}[Q]$	$P_{\mathrm{IB}}(W > 0)$, $E_{\mathrm{HB}}[C_{\mathrm{CB}}(W > 0)]$	$E_{\mathrm{HB}}[\%Ab]$, $E_{\mathrm{HB}}[C_{\mathrm{CB}}(Ab)]$		
	[248]	IS						W_t			
Other	[52]								E_{IB}[throughput]		
	[72]	SIPP						E_{HB}[Coverage], E_{IB}[Coverage]			
	[115]		x					$E_{\mathrm{HB}}[W]$	$E_{\mathrm{HB}}[\%Ab]$		$E_{\mathrm{HB}}[U]$
	[139]					x			E_{IB}[throughput]		
	[157]					x			E_{IB}[throughput]		
	[182]		x					$P_{\mathrm{HB}}(W \leq \tau)$			

*: assumed, not stated explicitly in article

Table 2.6: Classification by performance metrics

2.4.1 Stationary approximations

As Table 2.6 shows, stationary approximations are by far the most widely adopted approach for performance evaluation in time-varying systems. These approaches translate the nonstationary system parameters into stationary counterparts, which they feed into a (series of) stationary model(s). Various methods have been suggested; for detailed descriptions, we refer readers to Green et al. [99], Whitt [253] and Defraeye and Van Nieuwenhuyse [64]. Here, we limit ourselves to a brief discussion.

The Pointwise Stationary Approximation (PSA; [95, 91, 246]) uses the instantaneous arrival rate λ_t at each time t in a separate stationary model. The underlying assumption here is that the steady-state is realized almost immediately, which can be the case only if the number of arrivals and service completions per cycle is sufficiently high [246]. In a Stationary Independent Period-by-Period approach (SIPP, [96]), a separate stationary model instead gets applied to each discrete time interval, with the average arrival rate as the input parameter. Green et al. [96] present extensions to the SIPP approach, such as Lag SIPP, in which the arrival rate shifts by an amount of time proportional to the expected service time [70, 93]. This approach complies with the observation that in nonstationary systems, peaks in system congestion lag behind the arrival rate peaks [92, 230], as is commonly referred to using terms such as time lag or congestion lag. A lagged variant of PSA can be applied similarly [93]. Accounting for this lag can greatly improve the accuracy of SIPP (and PSA), particularly when the average service time —and thus the time lag—is long. Henderson and Mason [111] evaluate performance by applying a smoothing algorithm to the stationary results. This method, which serves as an improvement to the PSA approach, is capable of modeling the commonly observed congestion lag. Thompson [230] puts forward an Effective Arrival Rate approximation (EAR), that shifts the arrival rate proportional to the expected waiting time. Green and Kolesar [92] present the Simple Peak-Hour Approximation (SPHA), an approach that is popular in practice. SPHA approximates performance by a single stationary model, which takes the maximum arrival rate over the cycle as an input parameter.

The Modified Offered Load (MOL) approximations and infinite server (IS) approximations account for the congestion lag in a different way, by relying on analytically tractable results for infinite server queues [70, 71]. In IS, the time-varying number of customers N_t in the system is approximated by its infinite server counterpart N_t^∞ [131, 185] (e.g., the $M_t/G/s_t$ queue is approximated by an $M_t/G/\infty$ system). The delay probability, which can be obtained as $\Pr(N_t \geq s_t)$, is then approximated by $\Pr(N_t^\infty \geq s_t)$ [131]. In contrast, MOL entails a stationary approximation, such that at each moment in time, a stationary model gets applied, using the modified arrival rate $\lambda_t^{\text{MOL}} \equiv m_t^\infty \mu$, with m_t^∞ indicating the expected number of busy servers in an infinite-server system with the same arrival and service processes at time t. Details regarding MOL can be found in [130, 184, 61, 131, 185, 78]. Although the quantity m_t^∞ by definition disregards abandonments (as these do not occur in an infinite server system), MOL can be applied in systems with abandonments, by inserting λ_t^{MOL} in a (stationary) model with abandonments (Feldman et al. [78] report promising results for $M_t/M/s_t+M$ systems). Liu and Whitt [170] suggest the Delayed Infinite Server Offered Load (DIS-OL) method for staffing, an alternative offered load approach that targets overloaded systems and that is better tailored to performance metrics such as abandonment probability and expected waiting time. Hampshire et al. [107] extend the MOL approach to queues with limited capacity using the so-called fluid modified offered load, which provides insights into the number of blocked and abandoned customers.

The key advantage of stationary approximations lies in their simplicity: they can be applied to any system (regardless of the assumptions on service and abandonment processes, the priority rule, the system structure), as long as the stationary counterpart is available. However, the approach also has drawbacks. For instance, stationary approximations cannot be obtained in (temporarily) overloaded systems without abandonments, because the stationary system then is unstable. Their applicability and accuracy is also highly linked to the validity of the underlying assumptions, such as statistical independence of delays between separate intervals and steady-state being reached quickly in each interval [96]. Moreover, the stationary model itself may already be challenging, requiring the use of approximations (for ex-

ample, Whitt [249] and Iravani and Balcioğlu [124] provide approximations for the difficult $M/G/s+G$ queue). This may explain why many authors resort to the $M_t/M/s_t$ system, as closed-form results are available for the stationary $M/M/s$ queue [102]. Finally, the effect of the exhaustive service process cannot be accounted for with a stationary approximation, because the service policy is irrelevant in a stationary model.

Apart from the stationary approaches mentioned in Table 2.6, the literature also presents the Stationary Backlog-Carryover (SBC) approach [223, 222]. This approach does not appear in the categorization as it has not yet been used within an optimization framework for staffing or scheduling; it has been applied successfully to analyze time-dependent delays at airport runways and check-in counters though [224, 225]. The advantage of this approach is that, unlike other stationary approaches, it can be applied in temporarily overloaded systems without abandonments. Whereas most stationary approximations assume staffing intervals to be independent, SBC instead measures the "backlog" incurred in each period, and transfers it to the next period. As such, the link between the congestion of consecutive periods is captured.

2.4.2 Discrete-event simulation

As is evident from Table 2.6, discrete-event simulation is highly popular for performance evaluation. Discrete-event simulation can model complexities that go beyond the capabilities of analytical and numerical methods (see Law and Kelton [158] for a comprehensive textbook on discrete-event simulation). Especially in healthcare contexts, simulation is a widely adopted methodology (review articles include [104, 129, 138, 245]), but it also appears in other contexts, such as call centers [188]. Although simulation models are commonly context-specific (see, e.g., [5, 76, 87, 117, 187, 227] for applications of simulation in emergency departments), several efforts have sought to develop generic simulation models [80, 81, 197, 219, 103]. However, developing and validating a simulation model is often burdensome, and the computation time tends to be high.

2.4.3 Numerical methods

In an $M_t/M/s_t$ system, performance can be evaluated by numerically integrating the Ordinary Differential Equations (ODEs) that describe the system (see, e.g., Gross et al. [102] for general background; a more thorough description can be found in [156, 121, 94]). Several ODE-solvers, such as the Euler or Runge-Kutta ODE solver from the Matlab ODE Suite [213], seek to facilitate this analysis. Numerically solving ODEs offers a commonly used benchmark to assess the accuracy of stationary approximations [96, 97] or other methods.

Although Ingolfsson et al. [120] apply this approach, they also note that it requires substantial computational effort. A recent study by Ingolfsson et al. [121] compares several numerical performance evaluation methods in terms of their accuracy and speed for the $M_t/M/s_t$ system. They show that the randomization approach provides a level of accuracy similar to the ODE approach, at a substantially lower computational cost. Though randomization (or uniformization) originates in stationary queues [132, 90, 101], it can be applied successfully for personnel capacity planning in nonstationary queues too (as in Ingolfsson et al. [122]; see also Ingolfsson et al. [121], Ingolfsson [123] and Creemers et al. [57] for related work on performance evaluation in nonstationary queues using the randomization approach).

In general, both randomization and numerical solutions to ODEs rely heavily on Markovian assumptions. The majority of models use an exponential distribution for the service and/or abandonment process. Izady [126] describes how the methods can be extended to phase-type distributions, and concludes that the computational effort increases considerably (which is confirmed by the computational results in Creemers et al. [57]).

The numerical methods generally do not include abandonments or an exhaustive service policy. Ingolfsson [123] includes the exhaustive service policy in a randomization approach and outlines how abandonments can be accommodated. Creemers et al. [57] present a general randomization approach that includes abandonments, an exhaustive service policy and time-varying phase-type distributions for the service and abandonment processes.

None of the categorized articles use Discrete-Time Modeling (DTM,

[49, 37, 36, 242, 243, 126, 50]) or closure approximations [207, 54, 226, 238] with a view toward optimizing staffing or scheduling decisions; the available articles focus solely on performance evaluation. The advantage of DTM lies in its ability to accommodate general service time distributions, by approximating the service duration by a discrete process using two-moment matching (for further details we refer to [49, 37, 36, 242, 243]). Wall and Worthington [243] report distinct advantages over MOL and PSA, particularly when temporal overloading is present. However, the computational effort of DTM may be high [126] and the existing articles all study the $M_t/G/s$ system (i.e., no time-varying number of servers). Recently, Chassioti et al. [50] put forward a DTM approach for systems with abandonments; they focus on systems with low service level targets (i.e., long customer waiting times), where congestion may be affected greatly by abandonment behavior. Closure approximations appear to be less attractive: as Ingolfsson et al. [121] show, they are cumbersome to implement and dominated by other methods (e.g., MOL) in terms of both accuracy and computation speed.

2.4.4 Fluid models

Deterministic fluid models are intended for systems that do not display stochasticity, but can serve as approximations to derive time-dependent performance in stochastic systems. These methods rely on so-called "fluid scaling", such that the system gets scaled up (e.g., by multiplying arrival rates and the number of servers by the same factor), and the stochastic randomness accordingly decreases in importance, relative to system dynamics (see [109] for an example). Whitt [251] points out that fluid approximations are particularly useful to assess performance in systems that are temporarily overloaded, in which contexts many traditional methods fail (e.g., stationary approximations are no longer valid, because the assumed per period stationarity will result in an infinite queue). For underloaded systems, fluid approximations often fail to capture system dynamics accurately [10, 3, 133]. That is, because fluid models rely on approximating the stochastic system by its deterministic counterpart, they implicitly assume that queues will only start to build up if the traffic intensity exceeds 1 (hence

they target overloaded systems). Fluid models regard arrival and departure processes as continuous flows rather than discrete processes, and they tend to become more accurate as the number of servers grows large [251]. For additional literature on the use of fluid approximations for systems with exponential service and abandonment processes, we refer to Mandelbaum et al. [176, 177, 178, 179, 180], Ridley et al. [202], and Jiménez and Koole [133]. Other systems suggest general service and/or abandonment time distributions, including $G_t/G/s+G$ models (with state-dependent arrival rates, [250]), the $G_t/G/s_t+G$ model [171, 167, 168, 169], and networks of queues [166, 172]. Aguir et al. [3] apply fluid models to gain insight into a system with retrials. Personnel capacity planning methods also can rely on fluid models; existing studies [53, 108, 23, 24, 25, 28, 105] all focus on a setting with heterogeneous customers and servers and account for uncertainty in the arrival rate.

2.4.5 Empirical methods

Some authors rely on empirical data to estimate system performance. Nah and Kim [193] apply regression to express the abandonment percentage and the mean waiting time as a function of the arrival rate per server. The resulting expressions then are inserted in a mathematical program to obtain a minimum cost shift schedule. Lam et al. [157] and Kabak et al. [139] target shift scheduling in the retail sector. They rely on empirical data to link store sales with customer arrivals, staff number, and other factors. The staffing levels are then selected to maximize the expected profit. Andrews and Parsons [11], Quinn et al. [200], Lin et al. [164] include abandonment-related performance metrics in their models, that are derived from the service level by regression.

2.5 Classification by optimization approach

In this section, we classify previous publications according to the approach used to optimize personnel capacity. We make a distinction between staffing optimization (Section 2.5.1) and shift schedule optimization (Section 2.5.2).

2.5.1 Staffing approaches

Table 2.7 presents an overview of the different staffing methods. As is evident from this table, simple heuristics tend to be popular, such as the "Smallest Staffing Level" (SSL) approach and the Square-Root Staffing (SRS) rule. The SSL approach solves for the stationary model using different capacity values and selects the smallest staffing level that yields satisfactory performance. For example, the staff level s_t is selected by:

$$s_t = \arg\min\{c \in \mathbb{N} : \Pr(N_t \geq c) < \alpha\}\,, \tag{2.1}$$

if the performance target is to keep the delay probability $\Pr(W_t > 0) = \Pr(N_t \geq s_t)$ below a given target α, for each t. SSL requires an explicit evaluation of the performance metrics, which can be hard to obtain especially in more complex queueing systems for which closed-form results are not available (e.g., the $M_t/G/s_t + G$ queue; [170]). Accordingly, Table 2.7 reveals that many articles that resort to SSL ignore abandonments and assume exponential service times, such that the closed-form results for the $M/M/s$ queue are applicable:

$$\Pr(W > 0) = \frac{\left(\frac{(s\rho)^s}{s!}\right)\left(\frac{1}{1-\rho}\right)}{\left(\sum_{k=0}^{s-1}\frac{(s\rho)^k}{k!} + \left(\frac{(s\rho)^s}{s!}\right)\left(\frac{1}{1-\rho}\right)\right)}\,, \tag{2.2}$$

with ρ equal to λ/μ (this is the well-known Erlang-C formula).

The SRS rule does not explicitly evaluate the performance metrics. Instead, as a general rule-of-thumb, it sets capacity at time t equal to the offered load m_t, augmented by an amount of safety capacity that is proportional to the square root of the offered load:

$$s_t = m_t + \beta\sqrt{m_t}\,. \tag{2.3}$$

The safety factor β is related to the target delay probability α, and can be determined by inverting the Garnett or Halfin-Whitt delay function [88, 106], among others. Reducing the safety factor to zero results in *staffing to the offered load*; see [170]. The offered load that is inserted into the SRS formula depends on the performance approximation used: for instance, $m_t = m_t^{\infty}$

Staffing method	References	Objective	Performance constraints (targets are denoted as α_1, α_2, ...)	Kendall notation
SRS	[107]	Maximize profit (revenue per customer, penalty cost for positive waiting time, penalty cost for abandonments)	$E_{\mathrm{HB}}[\%Ab] < \alpha_1$ and $E_{\mathrm{HB}}[\%Bl] < \alpha_1$	$M_t/M/s_t + M$
	[131]	Allocate resources to meet target performance	$P(W_t > 0) \leq \alpha$ for all t	$G_t/G_t/s_t$
	[170]	Stabilize performance	$E_{\mathrm{TB}}[W] = \alpha_1$ for all t, $P_{\mathrm{TB}}(Ab) = \alpha_2$ for all t	$M_t/G/s_t + G$
	[248]	$W_t = 0$ for all arriving customers		$M_t/G/s_t$
	[258]	Allocate resources to meet target performance	$P_{\mathrm{IB}}(W > \tau) \leq \alpha$ for all staffing intervals	$M_t/M/s_t$
SSL	[44]	Provide lower bound on staffing requirements	One of the following performance constraints: $P_{\mathrm{TB}}(W > 0) \leq \alpha$ for all t, $P_{\mathrm{TB}}(W > \tau) \leq \alpha$ for all t, $E_{\mathrm{TB}}[W - \tau \vert W > \tau] \leq \alpha$ or $E_{\mathrm{HB}}[P_{\mathrm{TB}}(W > \tau)] \leq \alpha$	$G_t/G/s_t + G$
	[96]	Minimize labor cost	$P_{\mathrm{IB}}(W > 0) < \alpha$ for all staffing intervals	$M_t/M/s_t$
	[137]	Minimize labor cost	$P_{\mathrm{IB}}(W > \tau) \leq \alpha$ for all staffing intervals	$M_t/M/s_t + M$
	[170]	Stabilize performance	$E_{\mathrm{TB}}[W] = \alpha_1$ for all t, $P_{\mathrm{TB}}(Ab) = \alpha_2$ for all t	$M_t/G/s_t + G$
	[168]	Stabilize performance	$E_{\mathrm{TB}}[W] = \alpha_1$ for all t	$G_t/G/s_t + G$
	[200]	Minimize cost (labor, trunk lines, monthly rental costs)	-	$M_t/M/s_t + G$
Dynamic program	[82]	Minimize sum of (linear function of) labor cost and expected number in system		$G_t/M/s_t$
	[111]	Minimize labor cost	$E_{\mathrm{IB}}[CGOS] > \alpha$ for all staffing intervals	$M_t/M/s_t + M$
	[208]	Minimize cost (staffing, penalty if service level is not met)	Constraint on number available workspaces for staff	$M_t/M/s_t$
Mathematical programming	[23]	Minimize cost (staffing, abandonment, blocking, waiting)		$M_t/M/s_t + M$
	[24]	Minimize cost (staffing, abandonment, waiting)		$M_t/M/s + M$
	[25]	Minimize cost (staffing, penalty cost per abandonment)		$M_t/M/s + M$
	[28]	Minimize cost (staffing, waiting, abandonment)		$M_t/M/s_t + M$
	[31]	Minimize cost(cost of incurring positive delay, cost of hiring servers, cost of waiting, cost of using temporary servers)	$P_{\mathrm{HB}}(W = 0) > \alpha_1$, $E_{\mathrm{HB}}[Q] < \alpha_2$	$M_t/M/s$
	[105]	Minimize cost (staffing)	$P_{\mathrm{HB}}(Ab_t > \alpha) < \delta$	$M_t/M/s_t + M$
	[108]	Minimize cost (staffing, abandonment)		$M_t/M/s + M$
	[144]	Minimize labor cost	$Q_t < \alpha$ for all times t	$M_t/M/s_t + M$
Simulation-based heuristic	[5]	Maximize E_{HB}[throughput]	$E_{\mathrm{HB}}[W] < \alpha_1$, upper bound on staffing cost	$M_t/G/s$
	[55]	Minimize labor cost	$P_{\mathrm{IB}}(W > 0) \leq \alpha$ for all staffing intervals	$M_t/M/s_t$
	[65]	Minimize labor cost	$P_{\mathrm{TB}}(W > \tau) \leq \alpha$ for all time epochs t	$M_t/G/s_t + G$
	[78]	Stabilize performance	$P_{\mathrm{TB}}(W > 0) \leq \alpha$ for all time epochs t	$M_t/G/s_t + G$
	[143]	Minimize labor cost	$P_{\mathrm{IB}}(W \leq \tau) > \alpha$ for all staffing intervals	$M_t/M/s_t$

Table 2.7: Classification by staffing method

corresponds to an IS approximation, whereas $m_t = \bar{\lambda}_t/\mu$ complies with the SIPP method (with $\bar{\lambda}_t$ the average arrival rate over a given staffing interval). SRS can be applied as a simple heuristic to determine staffing levels in combination with either stationary approximations (e.g., SIPP, PSA, lagged SIPP or MOL [78]), as well as infinite server approximations [131], or in a network context [258]. The general background and applicability of SRS is provided in Gans et al. [85], Borst et al. [35], Whitt [247], Koole and Mandelbaum [152]. Although theoretical and empirical evidence in support of the SRS rule has grown [170, 78, 254], the main challenge in practical applications lies in determining the appropriate value for the safety factor [99, 170, 35, 88, 106].

Simulation-based heuristics use simulation as performance evaluation method in an iterative procedure, to guide the search process. They provide great flexibility in terms of system assumptions; they can be found in Feldman et al. [78], Defraeye and Van Nieuwenhuyse [65], Ahmed and Alkhamis [5], Corominas and Lusa [55], and Kim and Ha [143], among others. Feldman et al. [78] propose the promising iterative staffing algorithm (or ISA) for determining staffing requirements in $M_t/G/s_t + G$ queues, with a view toward stabilizing the delay probability. ISA repeatedly evaluates and alters the staffing function based on the distribution of the number in system at each time instant (which is estimated by simulation), until the desired performance is attained. Defraeye and Van Nieuwenhuyse [65] propose ISA(τ), addressing waiting time tail probabilities instead of delay probabilities. ISA(τ) updates the staffing vector based on the observed performance, multiplying the staffing levels with a factor proportional to the deviation from the performance target. Ahmed and Alkhamis [5] present a simulation-based heuristic that does not allow the staffing level to vary over time. As such, the dimension of the solution space remains limited to the number of resources available (6 in that study). Allowing staffing changes causes a steep increase in the dimension of the solution vector; we did not find applications of this type of approach for systems with a time-varying number of servers. Finally, Kim and Ha [143] and Corominas and Lusa [55] select staffing levels chronologically, on an interval-by-interval basis; their heuristics each time take the previously selected capacity levels (i.e., in ear-

lier staffing intervals) as given.

Most articles adopt a constraint-satisfaction approach, minimizing cost subject to one or more performance constraints that are commonly related to the quality of service (see, e.g., [65, 111]). For mathematical programming models though, the constraints are frequently included in the objective function by assigning a penalty cost (e.g., cost related to abandonments, blocking, waiting). An alternative objective is to pursue time-stable performance instead of minimizing costs [170, 78].

2.5.2 Shift schedule optimization

Table 2.8 classifies prior research according to its approach to shift schedule optimization. As discussed in Section 2.2, we distinguish three approaches: scheduling based on known staffing requirements (*two-step approach*), the *integrated approach*, and scheduling directly from demand (the *direct approach*).

According to Table 2.8, most articles adopt the *two-step approach*. This approach considers staffing and shift scheduling as separate, consecutive steps. The scheduling step then consists of finding the lowest cost schedule that meets a set of constraints, which are commonly related to work regulations (e.g., minimal amount of time between consecutive shifts, maximum number of working hours per week), employee preferences (e.g., full time versus part time) and covering the staffing requirements. Dantzig's set covering formulation [60] —though it dates back to the 50s— is still highly relevant and used frequently in the literature (see [122, 13, 14]). The staffing requirements are interpreted as strict constraints to be met in Dantzig's model. Alternatively, they can be seen as "desirable" levels that still allow for deviations, as proposed by Keith [140] (see [218, 125, 68], among others).

Shift optimization	Ref.	Objective	Constraints on quality of service	Kendall notation	Staffing optimization
Two-step approach					
Mathematical programming	[11]	Minimize cost (labor, waiting, lost calls)	-	$M_t/M/s_t + G$	SSL
	[32]	Minimize labor cost	$P_{\mathrm{IB}}(W < \tau) \geq \alpha$ for all staffing intervals	$M_t/M/s_t$	SSL
	[47]	Minimize labor cost	$P_{\mathrm{HB}}(LoS < \tau) \geq \alpha$	$M_t/G/s_t$	Simulation-based heuristic
	[52]	Minimize labor cost	$E_{\mathrm{IB}}[U] = \alpha$, for all staffing intervals	Not specified	Heuristic
	[68]	Minimize deviation between scheduled servers and staffing requirements	$P_{\mathrm{IB}}(W < \tau) \geq \alpha$ for all staffing intervals	$M_t/M/s_t + M$	SSL
	[72]	Maximize expected coverage	$\max E_{\mathrm{HB}}$[Coverage] and $maximin E_{\mathrm{IB}}$[Coverage]	$M_t/M/s_t$	Tabu search
	[75]	Minimize number of shifts	$P_{\mathrm{IB}}(W \leq \tau) > \alpha$ for all staffing intervals	$M_t/M/s_t$	SSL
	[115]	Minimize labor cost	$E_{\mathrm{HB}}[W] \leq \alpha$	$M_t/M/s_t + M$	Simulation-based heuristic
	[125]	Minimize penalty of over and understaffing w.r.t. staffing requirements	$P_{\mathrm{HB}}(LoS < \tau) \geq \alpha$	$M_t/G/s_t$	SRS
	[139]	Maximize (profit - staff cost)	-	Not specified	Nonlinear programming
	[161]	Minimize cost (labor understaffing w.r.t. staff requirements, overtime cost)	$P(P_{\mathrm{TB}}(W \leq \tau) < \alpha) < \delta$	$M_t/M/s_t$	SSL
	[162]	Minimize labor cost	$P(P_{\mathrm{TB}}(W \leq \tau) < \alpha) < \delta$	$M_t/M/s_t$	SSL
	[164]	Minimize overtime cost and cost of uneven manpower distribution	$E_{\mathrm{IB}}[\%Ab]$	$M_t/G/s_t + G$	SSL
	[182]	Minimize labor cost	$P_{\mathrm{HB}}(W \leq \tau)$, for all customers of a given flight	Not specified	Simulation-based heuristic
	[203]	Minimize cost (labor cost, penalty cost for not meeting aggregate service level constraint)	$P_{\mathrm{HB}}(W \leq \tau) > \alpha$	$M_t/M/s_t + M$	SSL
	[210]	Minimize cost (labor, waiting, abandonments)	$P_{\mathrm{IB}}(W > 0) \leq \alpha$	$M_t/M/s_t + M$	SSL
	[218]	Minimize penalty of over and understaffing wrt staffing requirements; Minimize $E_{\mathrm{HB}}[LoS]$	-	Not specified	SRS/offered load
	[230]	Minimize actually worked labor hours	$E_{\mathrm{HB}}[P_{\mathrm{IB}}(W \leq \tau)] \geq \alpha$	$M_t/M/s_t$	SSL
Mathematical programming + metaheuristic	[211]	Minimize cost (labor, waiting, abandonments)	$P_{\mathrm{IB}}(W > 0) \leq \alpha$	$M_t/M/s_t + M$	SSL

(continued on next page)

(continued from previous page)

Shift optimization	Ref.	Objective	Constraints on quality of service	Kendall notation	Staffing optimization
(Meta)heuristics	[83]	Maximize weighted scores w.r.t. performance targets	Various customer and server-based constraints (not specified)	Not specified	SSL
	[232]	1) Minimize labor cost 2) Maximize overall service, given total staffing	1) $P_{\mathrm{IB}}\{W < \tau\} > \alpha_1$ and $P_{\mathrm{HB}}\{W < \tau\} > \alpha_2$ 2) $P_{\mathrm{IB}}\{W < \tau\} > \alpha_1$	$M_t/M/s_t$	SSL
Trial-and-error	[2]	Minimize labor cost	$P_{\mathrm{HB}}(W \leq \tau) > \alpha$	$M_t/M/s_t$	SSL
Not specified	[98]	Reduce $E_{\mathrm{HB}}[\%Ab]$, allocate resources to meet target performance	$P_{\mathrm{IB}}(W > \tau) \leq \alpha$ for all staffing intervals	$M_t/M/s_t$	SSL
Integrated approach					
Mathematical programming	[13], [14]	Minimize labor cost	$P_{\mathrm{IB}}(W > \tau) \leq \alpha$ for all staffing intervals	$M_t/G/s_t$	
	[17]	Minimize labor cost	$P_{\mathrm{HB}}(W > \tau) \leq \alpha$; $P_{\mathrm{IB}}(W > \tau) \leq \alpha$ for all staffing intervals	$M_t/M/s_t$	
	[110]	Minimize labor cost	$E_{\mathrm{IB}}[CGOS] > \alpha$, for all staffing intervals	Not specified	
	[122]	Minimize labor cost	$P_{\mathrm{TB}}(W > \tau) \leq \alpha$ for all t	$M_t/M/s_t$	
Direct approach					
Free disposable hull analysis	[46]	Determine dominating schedules w.r.t. $E_{\mathrm{HB}}[W]$, $\max_{\mathrm{HB}}\{W\}$, $E_{\mathrm{HB}}[Q]$, $\max_{\mathrm{HB}}\{Q\}$, $E_{\mathrm{HB}}[\%Ab]$,$E_{\mathrm{HB}}[\%Bb]$, $P_{\mathrm{HB}}(W > \tau)$, $E_{\mathrm{HB}}[U]$	-	Not specified	-
Metaheuristic	[120]	Minimize labor cost	$P_{\mathrm{TB}}\{W > 0\} < \alpha_1$ (or alternatively, $P_{\mathrm{TB}}\{W > \tau\} < \alpha_2$)	$M_t/M/s_t$	-
Local search	[155]	Minimize number of shifts	$E_{\mathrm{HB}}[P_{\mathrm{IB}}(W > \tau)] \leq \alpha$	$M_t/M/s_t$	-
Mathematical programming	[86]	Minimize labor cost	$E_{\mathrm{HB}}[\%Ab] < \alpha_1$, $E_{\mathrm{IB}}[\%Ab] = \alpha_{i_s}$, for all staffing intervals i_s	$M_t/M/s_t + M$	-
	[109]	Maximize average profit	$E_{\mathrm{HB}}[W] < \alpha_1$, $E_{\mathrm{HB}}[\%\mathrm{Served}] > \alpha_2$	$M_t/M/s_t + M$	-
	[193]	Minimize cost (labor, waiting, abandonments)	$E_{\mathrm{IB}}[\%Ab] \leq \alpha_1$ and $E_{\mathrm{HB}}[\%Ab] \leq \alpha_2$	Not specified	-
	[157]	Maximize (profit - staff cost)	-	Not specified	-

Table 2.8: Classification by shift scheduling method

As noted in general overviews of the shift scheduling literature [232, 73, 74, 201, 39, 240], most analyses rely on mathematical programming techniques to find an optimal shift schedule. Search heuristics also can be used [232, 83]. Nearly all studies that adopt the two-step approach rely on either SSL or SRS to determine the staffing requirements. The two-step approach is appealing due to the difficulty of integrating stochastic performance constraints into mathematical models; with this approach, the performance constraints are taken care of in the staffing step, such that shift scheduling becomes a deterministic problem.

The *integrated approach* allows to determine staffing requirements and shift schedules simultaneously: it iteratively updates staffing requirements and fits the minimum cost shift schedule, until a satisfactory (not necessarily optimal) solution is found. One of the first integrated approaches can be found in Kolesar et al. [150], who combine SIPP and a mathematical model similar to Dantzig [60] to derive shift schedules for police patrol cars (we remark that the authors do not provide a systematic approach for updating the staffing requirements). Table 2.8 reveals that the dominant solution methodology in this case is mathematical programming. Henderson and Mason [110], Atlason et al. [13, 14] and Avramidis et al. [17] rely on cutting plane methods [141] to determine the optimal shift schedule and conduct simulations to evaluate system performance. Atlason et al. [13, 14] extend the work of Henderson and Mason [110]. The algorithm in Atlason et al. [13] requires the service level function to be a concave function of the staffing levels; however, because the service level function tends to follow an S-shaped curve as staffing increases [122, 14], the assumption has been relaxed toward pseudo-concavity in Atlason et al. [14]. Avramidis et al. [17] use a cutting plane method for simultaneous staffing and scheduling, and apply local search techniques to further improve the solution. Ingolfsson et al. [122] present a cutting plane algorithm that relies on randomization to evaluate performance. Campello and Ingolfsson [44] derive strict lower bounds on staffing (which are not necessarily feasible in conjunction with the performance constraint) and use them as a starting point in the algorithm of Ingolfsson et al. [122]. The integrated approach avoids the type of suboptimality that may arise in the two-step approach, as it determines

staffing requirements and shift schedules simultaneously. In that sense, it can be seen as superior to the two-step approach. Note, however, that the implementation of an integrated approach does not by definition guarantee that the obtained solution is optimal: for instance, the cutting plane algorithm in Ingolfsson et al. [122] may miss the optimum because the cuts are introduced based on *estimations* of the additional staffing that is required to meet the performance constraint. By contrast, Atlason et al. [13, 14] show that their method converges to the optimal solution as the number of replications in the simulation model grows large.

The *direct approach* does not rely on per-period staffing requirements; instead, it creates schedules directly from the arrival rates. It contains both heuristic and mathematical programming techniques. Ingolfsson et al. [120] use a genetic algorithm to generate schedules directly from demand, whereas Gans et al. [86] adopt a stochastic programming approach that takes forecasted arrival rates as an input. Surprisingly, Castillo et al. [46] are the only ones to treat the shift scheduling problem as a multi-criteria decision problem, by using free disposable hull analysis [239] to select a set of dominant schedules with respect to several performance metrics.

Each of the three approaches (two-step approach, integrated approach, and direct approach) has its own pros and cons. The two-step approach has the advantage of flexibility in the choice of the algorithms used in the separate staffing and scheduling steps. In spite of this flexibility, the majority of two-step approaches implement fairly basic scheduling models (e.g., similar to Dantzig's model). Although dedicated high-level scheduling algorithms (that are designed to account for realistic scheduling constraints in an efficient way) can easily be included in the two-step approach, we found no applications of this sort in the literature. The integrated approach and direct approach are both appealing because they avoid the type of suboptimality that may arise with the two-step approach. However, these models are often highly complex, implying that simplifications to the system assumptions may be required to keep the models solvable. The integrated approaches rely on staffing requirements to efficiently guide the search process, and their iterative nature allows to control the final schedule in terms of solution quality (iterations can be added until a satisfactory solution is

obtained). The direct approach is conceptually more straightforward (as it skips the staffing step), but the schedule optimization becomes more challenging because the solution space is less constrained.

2.6 Classification by application areas

Finally, Table 2.6 classifies articles on the basis of their application context. For each reference, we indicate whether the model was implemented (and the results reported), or if it was validated using real-life data or fictive examples. We only consider implementations reported in the academic literature and acknowledge that this is an incomplete indicator of practical implementation. For ease of reference, we repeat the methodology used for staffing and scheduling. As is evident from this table, emergency departments and call centers are the most popular (intended) application areas for the various types of models.

Context	Ref.	Implementation (+results reported) (Y/N)	Validation by means of real-life data (Y/N)	Validation by means of other (fictive) examples (Y/N)	Methodology staffing	Methodology scheduling
General	[44]	N	Y	Y	SSL	-
	[46]	N	Y	N	-	Free disposable hull analysis
	[78]	N	Y	Y	Simulation-based optimization	-
	[82]	N	N	N	Dynamic programming	-
	[96]	N	N	Y	SSL	-
	[120]	N	Y	N	-	Metaheuristic
	[122]	N	N	Y	Math. programming	id.
	[131]	N	Y	N	SRS	-
	[170]	N	N	Y	SRS; SSL	-
	[168]	N	N	N	SSL	-
	[230]	N	N	Y	SSL	Math. programming
	[232]	N	N	Y	SSL	Metaheuristic
Emergency department	[5]	N	Y	N	Simulation-based optimization	-
	[47]	N	Y	N	Simulation-based optimization	Math. programming
	[65]	N	Y	Y	Simulation-based optimization	-
	[98]	Y(+Y)	Y	N	SSL	Not specified
	[125]	N	Y	N	SRS	Math. programming

(continued on next page)

(continued from previous page)

Context	Ref.	Implementation (+results reported) (Y/N)	Validation by means of real-life data (Y/N)	Validation by means of other (fictive) examples (Y/N)	Methodology staffing	Methodology scheduling
	[218]	N	Y	N	SRS/offered load	Math. programming
	[258]	Y(+N)	Y	N	SRS	-
Call center	[2]	Y(+N)	Y	N	SSL	Trial-and-error
	[11]	N	Y	N	SSL	Math. programming
	[13]	N	N	Y	Math. programming	id.
	[14]	N	N	Y	Math. programming	id.
	[17]	N	Y	N	Math. programming	id.
	[23]	N	N	N	Math. programming	-
	[24]	N	N	Y	Math. programming	-
	[25]	N	N	Y	Math. programming	-
	[28]	N	Y	N	Math. programming	-
	[31]	N	N	Y	Math. programming	-
	[32]	N	Y	N	SSL	Math. programming
	[55]	N	N	Y	Simulation-based heuristic	-
	[68]	Y(+N)	Y	N	SSL	Math. programming
	[75]	N	Y	N	SSL	Math. programming
	[83]	Y(+Y)	Y	N	SSL	Set of heuristics
	[86]	N	Y	N	-	Math. programming
	[105]	N	N	Y	Math. programming	-
	[107]	N	N	Y	SRS	-
	[108]	N	N	Y	Math. programming	-
	[109]	N	N	Y	-	Math. programming
	[110]	N	N	N	Math. programming	id.
	[111]	N	Y	N	Dynamic program	-
	[137]	N	Y	N	SSL	-
	[143]	N	Y	N	Simulation-based heuristic	-
	[144]	N	Y	N	Math. programming	-
	[155]	N	Y	N	-	Local search
	[161]	N	Y	N	SSL	Math. programming
	[162]	N	Y	N	SSL	Math. programming
	[164]	N	Y	N	SSL	Math. programming
	[193]	N	Y	N	-	Math. programming
	[200]	Y(+Y)	Y	N	SSL	-
	[203]	N	Y	N	SSL	Math. programming
	[208]	N	Y	Y	Dynamic programming	-
	[211]	N	Y	N	SSL	Math. programming + metaheuristic
	[210]	N	Y	N	SSL	Math. programming
	[248]	N	N	N	SRS	-
Other	[72]	N	Y	N	Metaheuristic	Math. programming
	[182]	Y(+Y)	Y	N	Simulation-based heuristic	Math. programming
	[52]	Y(+Y)	Y	N	Heuristic	Math. programming
	[115]	Y(+Y)	Y	N	Simulation-based heuristic	Math. programming
	[139]	N	Y	N	Math. programming	Math. programming
	[157]	N	Y	N	-	Nonlinear programming

Table 2.9: Classification by real-life application

Within the set of articles we consider, Quinn et al. [200], Fukunaga et al. [83], Green et al. [98], Mason et al. [182], Hueter and Swart [115] and Choi et al. [52] are the only studies to implement a model and report the results; they all rely on the two-step approach. Quinn et al. [200], Fukunaga et al. [83] and Green et al. [98] used the (relatively unsophisticated) SSL approach to set staffing levels. Fukunaga et al. [83] also rely on SSL and complement their analysis with various search heuristics designed to select optimal shift schedules (however, they remain rather vague on the details of the proposed staffing and scheduling algorithms). Quinn et al. [200] apply a profit-driven approach, where the performance target is included in the objective such that personnel is added as long as the incremental cost does not exceed the additional revenue (a similar logic can be found in Lam et al. [157] and Kabak et al. [139], in a retail setting). Mason et al. [182] and Hueter and Swart [115] apply a simulation-based heuristic (for staffing) and mathematical programming (for scheduling). Choi et al. [52] set staffing levels based on a heuristic and further refine the schedule using mathematical programming.

The implementations of staffing and scheduling models resulted in, among others, higher revenues [200], reductions in the labor cost [115, 52], less abandoned customers [200, 83, 98], better service levels and lower average waiting times Quinn et al. [200]. However, Mason et al. [182] highlighted that the initial schedules provided by their algorithm needed adjustments, because inadequate forecasts had caused understaffing. Moreover, they reported increases in the level of sick leave, possibly caused by the higher complexity of the new schedules.

Zeltyn et al. [258], Dietz [68], and Agnihothri and Taylor [2] assert that their models were implemented, but they do not provide any results about the actual implementation. Instead, they use real-life data to validate the model. The remainder of the publications lack any real-life implementation, though the large majority provide a model validation using real-life data or other means. Henderson and Mason [110], Whitt [248], Bassamboo and Zeevi [23], and Liu and Whitt [168] did not provide any type of implementation or validation for the proposed staffing and/or scheduling model.

Note that not only the objectives and definition of quality of service differ between the application contexts, but also the data availability. Detailed data are often readily available in call centers – this is in general not the case in retail stores and emergency departments. Lam et al. [157] report that the models put forward in the retail literature take advantage of the data that *is* available, for instance, using sales data to construct schedules. Moreover, practical implementations may not find their way to the academic literature due to data confidentiality.

We observe a trend toward models that place a greater emphasis on practical applicability. Dietz [68] provides a spreadsheet-based scheduling approach that can easily be used by practitioners; Gans et al. [86] present an integrated approach for forecasting, staffing and scheduling under parameter uncertainty; and Sinreich and Jabali [218] and Izady and Worthington [125] both rely on a generic simulation model for staffing and scheduling in an emergency department. Nah and Kim [193] use regression analysis to link waiting times to the observed offered load in a call center, as such they avoid using (often complex) queueing models. Lin et al. [164] apply a similar approach, but resort to stationary approximations in those staffing intervals where the regression model's performance is insufficient.

2.7 Conclusions and future research

The extensive review of extant literature we have reported leads us to draw several conclusions that may be useful for guiding further research. First, it becomes clear that this research field is growing rapidly. Researchers have grown very creative in applying multiple methodologies to optimize staffing and/or scheduling in systems with nonstationary demand, and thus meet a myriad of objectives and performance constraints. Unfortunately, our analysis of the system assumptions in Section 2.3 reveals that, all too often, their ambitious models still rely on rather theoretical assumptions (e.g., homogeneity of customers and servers, exponential assumptions for service and abandonment processes, single-stage systems). This reliance may help to explain why, as we observed in Section 2.6, many models lack a real-life implementation (and the few articles that report on real-life implementation

appear limited to relatively simple stationary approximations). In particular, only few contributions have tried to tackle a network setting with general service processes ([218, 125, 5], and presumably [83] too), but none of them has addressed general abandonment times in a network —despite the seemingly high relevance of this topic in many practical situations.

Correspondingly, as we observed in Section 2.4, stationary approximations remain highly popular as a performance evaluation method; in recent years, fluid models have also increased in importance (in particular in the call center literature). Both methodologies often rely on rather strict assumptions that may limit their applicability in practice though. We emphasize that the attractiveness of stationary approximations lies not in their accuracy but rather in their ability to provide a simple means to obtain rough guidelines of system performance. The obtained staff levels could be improved further on the basis of, for instance, a simulation model (as in Ertogral and Bamuqabel [75], Zeltyn et al. [258], Izady and Worthington [125]). In fact, hybrid methods that combine the simplicity and insights of queueing results with the flexibility and accuracy of simulation, provide great opportunities for analyzing highly complex settings. Our analysis revealed that authors tend to stick to exponential assumptions for their model description and validation (e.g., [258]), even when using simulation-based methods that are in principle readily extendable to general assumptions.

Our analysis in Section 2.4 also reveals the wide range of performance metrics being used in current research. It is intuitively clear that logical links exist among the different metrics (e.g., waiting time-related performance metrics relate to abandonment metrics and length-of-stay metrics). Surprisingly though, we were unable to find a single publication that explicitly aimed to uncover these links in complex settings (e.g., network settings with nonstationary arrivals, general service and abandonment times). Further examination of the relationships across different performance metrics in complex nonstationary systems may open up interesting opportunities for continued research, especially in relation to performance metrics that are difficult to compute. The discovery of an easy-to-compute proxy metric can then substantially simplify the performance evaluation phase and may often be sufficient to guide the search for adequate personnel schedules (as

in Izady and Worthington [125] and Green et al. [98]).

We found that some promising performance evaluation methods (e.g., the SBC or DTM approaches) have not yet found their way to staffing and scheduling algorithms; instead, the algorithms tend to resort to those methods that are the most common or straightforward (e.g., stationary approximations such as SIPP or PSA). A challenging direction for future research consists in achieving a better connection between the research fields on performance evaluation on the one hand, and staffing and scheduling on the other hand. However, performance evaluation should be well-aligned with the optimization methodology, especially in terms of computational requirements (computationally expensive evaluation methods ideally require optimization algorithms that quickly find a good solution).

The applicability of the models extends beyond the typical contexts presented in academic research (i.e., call centers and healthcare systems) to other settings, such as queues in retail stores, restaurants and banking. Care should be taken, though, because these systems tend to function on a much smaller scale. Consequently, adding or removing a single server can cause drastic changes in performance. Current literature does not devote much attention to this inherent discreteness of capacity or its implications for model performance (e.g., Feldman et al. [78] report weak performance of the ISA algorithm in case arrival rates are extremely low). The further development of models and algorithms that specifically target small-scale systems provides a promising avenue for further research.

Moreover, the use of the models need not necessarily be restricted to personnel planning; they are in fact relevant to a broad range of problem settings. Zhang et al. [259] provide an interesting illustration: the authors determine the year-by-year capacity of beds in a hospital by a simulation-optimization algorithm that is similar to the method of Defraeye and Van Nieuwenhuyse [65], and compare the results with MOL and SIPP.

Finally, to limit the scope of this study, we did not elaborate on demand forecasting or rostering. Inaccurate forecasts may cause inadequate staffing and scheduling (see for instance the implementation results in Mason et al. [182]). This can be accommodated by including parameter uncertainty in the model (see also Section 2.3); the integrated approach for forecasting, staffing,

and scheduling with parameter uncertainty of Gans et al. [86] represents an important step towards achieving a closer integration of the different phases in the capacity planning process. Our research reveals that staffing and scheduling for systems with nonstationary demand currently do not tend to integrate the rostering step. Though it can be expected that complexity will increase severely, including the rostering step in an integrated approach is likely to be valuable to avoid suboptimality. Moreover, various demand management strategies can be applied to manipulate the arrival rate itself, which in turn may facilitate the capacity planning process (as discussed in Chapter 1).

Chapter 3

Computing the probability of excessive waiting in $M_t/G/s_t + G$ queues with an exhaustive service policy

3.1 Introduction

In this chapter, we compare multiple methods used to compute the probability of excessive waiting (i.e., the probability that the waiting time exceeds a given threshold τ) in an $M_t/G/s_t + G$ queue with an exhaustive service policy. The main evaluation criteria are accuracy (for several values of τ) and computational cost. Performance evaluations for such systems are complicated, because (1) the customer arrival rate fluctuates over time, (2) customers may abandon the queue before receiving service, (3) service and abandonment processes are generally distributed, and (4) the exhaustive policy implies that servers work overtime if service is ongoing at the time that the server is scheduled to go off duty. These four characteristics are highly relevant in many real-life service systems. Current literature focuses mainly on $M_t/M/s_t + M$ systems, yet it has been shown that service and abandonment times are not always Markovian in practice [38]. Prior liter-

ature also frequently presumes a preemptive service policy: the customer is sent back to the queue if a server is scheduled to leave, because server overtime is not allowed [121, 51].

This chapter is inspired by work by Ingolfsson et al. [121], who compare several performance evaluation methods for $M_t/M/s_t$ queues with an exhaustive service policy. Their investigation includes numerical solution of the ordinary differential equations (ODEs), closure approximations, direct infinite-server (IS) approximations, modified offered load (MOL) approximations, effective arrival rate (EAR) approximations (as suggested by Thompson [230]), and the lagged stationary independent period-by-period (lagged SIPP) approximation. They conclude that randomization yields very accurate results, at a substantially lower computational cost than is required to solve the ODEs numerically. The other heuristics (MOL, EAR, Lagged SIPP, IS) perform worse in terms of accuracy but require significantly less computation time than does randomization. Among these fast heuristics, MOL tends to be the most accurate. Unlike Ingolfsson et al. [121], we consider general service times and include a general customer abandonment process. We thus focus on methods that do not rely on Markovian assumptions, which leads us to select the following methods: (1) discrete-event simulation; (2) the MOL approximation, which can apply to non-Markovian service and abandonment processes [99, 78]; and (3) the randomization method suggested by Creemers et al. [57]. To the best of our knowledge, no results are available on the performance of MOL in an $M_t/G/s_t + G$ queue; all previously reported computational results have been obtained using exponentially distributed service and abandonment times [78, 131, 130]. The numerical approximation suggested by Creemers et al. [57] aims at $G_t/G_t/s_t + G_t$ queues with an exhaustive service policy. Whereas they only report computational results for queue size, we address the probabilities of excessive waiting. We explore whether this (generally more time-consuming) approach provides distinct advantages over simulation or the MOL approximation.

We focus on small- to medium-scale systems (i.e., average offered load up to 10, average arrival rates up to 60 customers per hour, and an average utilization up to 95%), which appear in many practical applications (e.g., banks, retail stores, small call centers, emergency departments). By per-

forming a large computational experiment, we can compare several methods that provide the probability of excessive waiting in $M_t/G/s_t + G$ queues with an exhaustive service policy.

In Section 3.2, we introduce the notations used in this chapter. The methods studied in this chapter are described in Section 3.3. Section 5.5 details the computational experiment; we compare the methods in terms of accuracy (Section 3.4.3) and computational cost (Section 3.4.4). In Section 3.5, we highlight our key findings and suggest directions for further research.

3.2 Notation

We focus on the single-stage multiserver $M_t/G/s_t + G$ queue, with a first-in first-out queueing policy. Customers enter the system according to a cyclic Poisson pattern with a time-varying arrival rate λ_t over the time horizon $[0, T]$ (e.g., day, week, month). The current time is represented by $t \in [0, T]$. Service times and customer impatience times follow a general distribution: μ denotes the per-server service rate, and θ represents the abandonment rate (both are constant over $[0, T]$).

To evaluate performance, the time horizon $[0, T]$ is divided into intervals of Δ_p time units. The number of performance intervals in $[0, T]$ is denoted $I_p \equiv T/\Delta_p$ and $\mathbf{I}_p = \{1, \ldots, I_p\}$ represents the set of performance interval indices. The starting times of each performance interval $i_p \in \mathbf{I}_p$ are contained in $\mathbf{t}_p = \{0, \Delta_p, 2\Delta_p, \ldots, T - \Delta_p\}$.

We distinguish between *virtual* waiting times and *observed* waiting times. The virtual waiting time W_t is defined as the waiting time encountered by a virtual customer arriving at time $t \in \mathbf{t}_p$ [102, 159, 174]. In contrast, the observed waiting time W_{i_p} represents the waiting time experienced by actual customers arriving in the interval $i_p \in \mathbf{I}_p$. Virtual customers have infinite patience (i.e., they do not abandon the queue), whereas real customers have finite patience and may abandon the queue, if their wait grows large. We define τ as the maximum allowed waiting time, such that it provides a threshold value used to distinguish between acceptable waiting times (i.e., W_t lower than or equal to τ) and excessive waiting times (i.e.,

W_t exceeds τ).

Capacity changes can occur only at specific points in time. The *staffing interval*, with length Δ_s, denotes the interval over which capacity remains constant (we assume that Δ_s is an exact multiple of Δ_p). We assume $I_s \equiv T/\Delta_s$ staffing intervals in $[0, T]$; the set of staffing interval indices is represented as $\mathbf{I}_s = \{1, \ldots, I_s\}$. In addition, $\mathbf{t}_s = \{0, \Delta_s, 2\Delta_s, \ldots, T - \Delta_s\}$ represents the set containing the staffing interval start times, for all $i_s \in \mathbf{I}_s$. As such, the capacity at time t is given by:

$$s_{(t|t\in[t_s,t_s+\Delta_s[)} = s_{i_s} \ ,$$

with t_s the start time of staffing interval i_s. The different types of intervals and respective notations are illustrated in Figure 3.1.

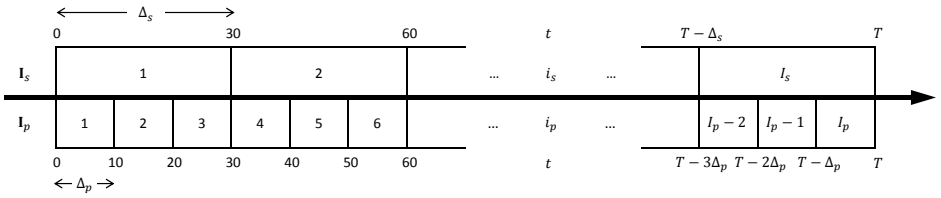


Figure 3.1: Illustration of staffing intervals and performance intervals.

3.3 Computational methods

Prior literature offers multiple approaches to evaluate performance in queues with a time-varying arrival rate and a time-varying number of servers. The most common approaches are summarized in Chapter 2 (see Section 2.4). With our focus on performance evaluation in $M_t/G/s_t + G$ queues, we chose to compare discrete-event simulation, the MOL approach, and the randomization approach suggested by Creemers et al. [57].

We distinguish between the *virtual* probability of excessive waiting calculated at times $t \in \mathbf{t}_p$ and the *observed* probability determined over intervals $i_p \in \mathbf{I}_p$. In doing so, we present three types of approaches: Sections 3.3.1 and 3.3.2 outline simulation-based approaches (referred to as SIM-VIRT, SIM-OAM, and SIM-OWM), whereas Section 3.3.3 elaborates on the

MOL approximation. Then in Section 3.3.4, we discuss the numerical approach suggested by Creemers et al. [57] (G-RAND). As shown in Table 3.1, SIM-VIRT and G-RAND attempt to evaluate virtual waiting times, whereas SIM-OAM, SIM-OWM, and MOL address observed waiting times.

Type	Method
Virtual waiting times	SIM-VIRT
	G-RAND
Observed waiting times	SIM-OAM
	SIM-OWM
	MOL

Table 3.1: Overview of methods

3.3.1 Simulation of virtual waiting times

The virtual waiting time corresponds to the time between t and the earliest time at which a (scheduled) server becomes available, because all customers that arrived before t have been served [102, 159, 174]:

$$W_t = \min\{w : \left(N^t_{t+w} \leq s_{t+w} - 1\right) \wedge (w \geq 0)\},$$

with s_{t+w} the capacity at time $t+w$ and N^t_{t+w} the number of customers arrived before time t that are still in system at time $t + w$. Note that the virtual waiting time is measured at a particular time instant (as opposed to *observed* waits, which are measured over an interval). The virtual waiting time distribution can be measured in a straightforward way through simulation. We insert a virtual (dummy) customer into the system at each time $t \in \mathbf{t}_p$ in replication r, such that the virtual waiting time $W_{t,r}$ equals the time at which this dummy customer would enter service. Let R represent the total number of replications in the simulation run. Define $\delta_{t,r}$ as a binary variable that signals whether the virtual waiting time exceeds the target τ for a given time t and replication r:

$$\delta_{t,r} = \begin{cases} 1 & \text{if } W_{t,r} > \tau, \\ 0 & \text{otherwise .} \end{cases}$$

The probability of excessive waiting at time t then can be estimated as:

$$\text{SIM-VIRT:} \qquad \Pr(W_t > \tau)_{\text{SIM-VIRT}} = \frac{1}{R} \sum_{r=1}^{R} \delta_{t,r} \, .$$

3.3.2 Simulation of observed waiting times

The time-dependent probability of excessive waiting can be measured from the *actual* or observed waiting times (e.g., Atlason et al. [14]). Waiting times are then aggregated over all customers that arrive during the interval $i_p \in \mathbf{I}_p$. The observed waiting time W_{i_p} typically differs from the virtual waiting time W_t, even if Δ_p becomes infinitesimally small, because virtual customers have infinite patience (by convention). Observed waiting times may be lower, due to customer abandonment. The magnitude of the difference depends on τ and θ: it decreases as the expected time to abandon (i.e., θ^{-1}) grows larger, compared with τ. As τ or θ approaches 0, the difference between virtual and observed waiting time disappears.

The observed probability of excessive waiting over a given interval i_p can be estimated by its arithmetic mean:

$$\text{SIM-OAM:} \qquad \Pr(W_{i_p} > \tau)_{\text{SIM-OAM}} = \frac{1}{R} \sum_{r=1}^{R} \left(\frac{\mathcal{L}_{i_p,r}}{\mathcal{A}_{i_p,r}} \right) , \qquad (3.1)$$

where $\mathcal{A}_{i_p,r}$ represents the number of arrivals during interval i_p of replication r, and $\mathcal{L}_{i_p,r}$ represents the number of customers experiencing a waiting time longer than τ. The probability that no arrivals occur in a given interval i_p increases as Δ_p decreases. Consequently, the number of replications needed to obtain accurate estimations with this approach increases as Δ_p decreases.

Atlason et al. [14] state that Equation 3.1 may be misleading, because it grants too much weight to sample realizations with low arrival volumes. They recommend the weighted mean SIM-OWM, in which each observation

of $(\mathcal{L}_{i_p,r}/\mathcal{A}_{i_p,r})$ is weighted with the realized number of arrivals:

$$\begin{aligned} \text{SIM-OWM}: \qquad \Pr(W_{i_p} > \tau)_{\text{SIM-OWM}} &= \frac{E[\mathcal{L}_{i_p}]}{E[\mathcal{A}_{i_p}]} \\ &\approx \sum_{r=1}^{R} \frac{\mathcal{L}_r(i_p)}{\mathcal{A}_{i_p,r}} \frac{\mathcal{A}_{i_p,r}}{\sum_{r=1}^{R} \mathcal{A}_{i_p,r}} \\ &= \frac{\sum_{r=1}^{R} \mathcal{L}_{i_p,r}}{\sum_{r=1}^{R} \mathcal{A}_{i_p,r}}. \end{aligned} \tag{3.2}$$

As detailed in Section 3.4.3, we often observe that $\Pr(W(i_p) > \tau)_{\text{SIM-OWM}} > \Pr(W(i_p) > \tau)_{\text{SIM-OAM}}$ (in line with the findings of Maman [173]). Maman [173] distinguishes between short-term and long-term performance measures: the arithmetic average (Equation 3.1) is acceptable when evaluating performance over short intervals of time, but Equation 3.2 should apply to longer intervals, because poor performance affects more customers in such an interval (with a greater emphasis on intervals with a high arrival rate). A key issue is the number of replications needed to accurately estimate the probability of experiencing an excessive waiting time, as we discuss in Section 3.4.4. In the computational experiment, the focus lies on SIM-OWM, though we briefly elaborate on SIM-OAM in Section 3.4.3, to illustrate the difference between both methods.

3.3.3 The MOL approximation

The modified offered load approach uses a stationary $M/G/s + G$ queuing model to approximate the performance of the $M_t/G/s_t + G$ queue at any given time t. The arrival rate of the stationary model can be obtained using the analytically tractable results for infinite server queues [70, 71]. Let m_t^{∞} be the expected number of busy servers in the corresponding $M_t/G/\infty$ system (equal to the offered load, in this case), and let $G_{(\cdot)}$ represent the service time cumulative distribution function. The modified arrival rate then

can be expressed as [99, 78]:

$$\lambda_t^{\text{MOL}} \equiv m_t^{\infty}\mu\,,$$
$$\text{where } m_t^{\infty} = \int_{-\infty}^{t} [1 - G_{(t-u)}]\lambda_u \,\mathrm{d}u\,.$$

The probability of excessive waiting at an arbitrary time t, denoted $\Pr(W_t > \tau)$, then can be derived as the excess wait probability resulting from the stationary $M/G/s+G$ queue with capacity s_t and arrival rate λ_t^{MOL}. Obtaining the stationary probability of excessive waiting for the $M/G/s+G$ queue is challenging. For this chapter, we determine it by means of simulation; alternatively, dedicated approximations for the $M/G/s+G$ queue could be used (see for instance Whitt [249] and Iravani and Balcioğlu [124]).

In our experiment, we center on *observed* waiting times for the the MOL approximation and use simulation to estimate the observed probability of excessive waiting $\Pr(W_{i_p} > \tau)_{\text{MOL}}$ for each $i_p \in I_p$. Let $\bar{\lambda}_{i_p}^{\text{MOL}}$ denote the average of λ_t^{MOL} over interval i_p. The probability of excessive waiting over interval i_p is estimated as the probability of excessive waiting in the stationary $M/G/s+G$ queue with capacity s_{i_p} and arrival rate $\bar{\lambda}_{i_p}^{\text{MOL}}$. We use SIM-OWM (i.e., the weighted average), but SIM-OAM (i.e., the arithmetic average) is applicable too. To ensure that the estimated probabilities of excessive waiting are sufficiently accurate, we add replications until the confidence interval halfwidths are smaller than 0.01 (for all τ). Note that MOL is unable to account for the effect of the exhaustive service process, because the service policy is irrelevant in a stationary model.

3.3.4 Randomization for $M_t/G/s_t+G$ queues

Creemers et al. [57] provide a performance evaluation method for $G_t/G_t/s_t+G_t$ queues with an exhaustive service policy. In the remainder of this chapter, we refer to this method as G-RAND. The approach builds on the work of Ingolfsson et al. [121], who extend the randomization approach of Grassmann [90] to time-varying $M_t/M/s_t$ queues with exhaustive service. Creemers et al. [57] further extend the randomization approach, by implementing abandonments and adapting the method to apply in settings with general

service, abandonment, and arrival times. The general distributions are approximated by continuous-time phase-type distributions, that decompose the general distribution into a set of exponential building blocks (so-called phases). The state of the system (i.e., number of customers in the queue) gets evaluated at discrete moments in time. In between these discrete moments, the arrival rate is assumed to be constant. The time between two observation moments is denoted Δ_g, where $\Delta_g \leq \Delta_p$. If Δ_g is sufficiently small, the transient distribution of the number of customers in the system, N_t, may be accurately obtained for all $t \in \mathbf{t}_p$. The model can be used to compute the virtual waiting time distribution at any time t by means of a death process, but doing so demands considerable computational effort, especially if system performance needs to be evaluated frequently (i.e., if Δ_p is small). The accuracy of G-RAND depends on the choice of Δ_g: lower values result in higher accuracy but also require more computation time. Moreover, the required computation time increases with the maximum queue size and maximum number of servers; therefore, this method mainly targets small- to medium-scale systems.

3.4 Computational experiment

We use a simulation study to assess the accuracy and computational cost of the methods described in the previous section (SIM-VIRT, SIM-OWM, MOL, and G-RAND) for a set of 162 problem instances. All methods are implemented in Visual Studio C++. The experiments are performed on an Intel I7 3.40 GHz computer with 8 GB RAM. The experimental setup and performance metrics are described in Sections 3.4.1 and 3.4.2, respectively. Section 3.4.3 compares the methods in terms of accuracy; the key differences are further explored by means of an illustrative example. Section 3.4.4 explores the trade-off between computational cost and model accuracy.

3.4.1 Experimental setting

Table 3.2 provides an overview of the parameter settings we used to construct the test set. The parameters give rise to 162 problem instances, which are

representative of small- to medium-scale systems. The time horizon equals one day (i.e., 1440 minutes), divided into smaller periods of length Δ_p.

Parameter	Symbol	Values
Time horizon (min)	T	1440
Staffing interval length (min)	Δ_s	30
Performance interval length (min)	Δ_p	10
Relative amplitude (arrival and staffing)	RA	0.5
Service rate (customers/hour)	μ	$\{1, 2, 6\}$
Abandonment rate (customers/hour)	θ	$\{0.5\mu, 2\mu\}$
Average capacity	$\bar{s}$	$\{2, 5, 10\}$
Average traffic intensity	$\bar{\rho}$	$\{0.5, 0.75, 0.95\}$
Squared coefficient of variation (service and abandonment)	C^2	$\{0.5, 1, 2\}$
Maximum allowed waiting time (min)	τ	$\{0, 10, 20\}$
Number of replications (in SIM-VIRT, SIM-OWM)	R	$\{250, 500, 1000, 2000, 4000, 8000\}$
Granularity used in G-RAND (min)	Δ_g	$\{0.125, 0.25, 0.5, 1, 2\}$

Table 3.2: Parameter settings in the computational experiment.

The time-varying arrival rate λ_t is modeled as a sine function with cycle equal to T (as in Feldman et al. [78], Ingolfsson et al. [121], among others). The relative amplitude (RA) is given by $\text{RA} \equiv A/\bar{\lambda}$, with A as the absolute amplitude and $\bar{\lambda}$ as the average arrival rate over the time horizon. More formally:

$$\lambda_t = \bar{\lambda}\left(1 + \text{RA}\sin\left(\frac{2\pi t}{T}\right)\right).$$

Note that $\bar{\lambda}$ is determined uniquely by the average capacity $\bar{s}$, the service rate μ, and the average traffic intensity $\bar{\rho} \equiv \bar{\lambda}/(\bar{s}\mu)$. The applicability of the models does not depend on the use of this sine function. Given the parameter settings in Table 3.2, it follows that $\bar{\lambda}$ ranges between 1 and 57 customers per hour. To limit the size of the test set, we assume that the service and abandonment processes have the same C^2 (i.e., 0.5, 1, or 2).

The staffing process is modeled using a sine function with relative amplitude equal to that of the arrival process (0.5 in the experiment). The staffing level in a given staffing interval is determined as the mean of this

sine function at the start and the end of the interval:

$$s_{i_s} = \bar{s} + \frac{1}{2}\mathrm{RA}\left(\sin\left(\frac{2\pi t}{T}\right) + \sin\left(\frac{2\pi\left(t+\Delta_s\right)}{T}\right)\right), \text{ with } t \in \mathbf{t}_s\,.$$

As is evident from Table 3.2, staffing intervals span 30 minutes.

In addition, G-RAND requires the arrival rate function to be piece-wise constant; we used 5-minute intervals, where the arrival rate is calculated as the mean of the arrival rate at the start and the end of the interval. For $C^2 < 1$, the service and abandonment distributions are modeled using a hypo-exponential distribution. If $C^2 = 0.5$ (i.e., the value used in our experiment), it is equivalent to an Erlang distribution with two phases. The respective parameters appear in Table 3.3 (a general discussion of how we obtained them is given in Appendix A). If $C^2 > 1$, we would use a two-phase Coxian distribution. The first phase has an exponential rate equal to $\mu\kappa^{-1}$, where μ^{-1} is the process mean and κ is a weighing factor [57]. The second phase is visited with probability β and has an exponential rate μ_2. For $C^2 = 1$, we use a single phase that has an exponentially distributed duration. The distributions are implemented likewise in the simulation-based approaches. As we show in Appendix A, these distributions yield closed-form expressions for the infinite server offered load, which is required for MOL.

	Erlang ($C^2 = 0.5$)				Two-phase Coxian ($C^2 = 2$)		
	$\mu = 1$	$\mu = 2$	$\mu = 6$		$\mu = 1$	$\mu = 2$	$\mu = 6$
μ_1	2	4	12	μ_1	2	4	12
μ_2	2	4	12	μ_2	0.5	1	3
				β	0.25	0.25	0.25

Table 3.3: Distribution parameters.

The exhaustive service policy in SIM-VIRT and SIM-OWM is modeled as in G-RAND: if capacity decreases, idle servers leave first, and then the remaining number of (randomly selected) busy servers starts to work overtime. The service policy is irrelevant in MOL, because this approach relies on stationary models.

3.4.2 Performance metrics

Let $\Pr\left(W(\cdot) > \tau\right)_{\text{TRUE}}$ denote the "true" value of the probability of excessive waiting (where $(\cdot)$ refers to either a moment in time $t \in \mathbf{t}_p$ *or* an interval $i_p \in \mathbf{I}_p$), as determined by means of a highly accurate (and thus computationally expensive) simulation model. We apply separate simulation models to obtain the true values for the observed and virtual metrics; each simulation uses 1,000,000 replications (the CPU time amounted to 30 minutes, on average).

We define $\Pr\left(W_{(\cdot)} > \tau\right)_{\text{EST}}$ as the probability of excessive waiting estimated using any of the methods described in Section 3.3 (i.e., focussing on virtual or observed waiting times). Formally, the absolute error is given by:

$$\text{AE}(\cdot) = \left|\Pr\left(W_{(\cdot)} > \tau\right)_{\text{TRUE}} - \Pr\left(W_{(\cdot)} > \tau\right)_{\text{EST}}\right|.$$

The mean absolute error (MAE) yields an aggregate performance metric per problem instance, obtained by taking the time-average of $\text{AE}(\cdot)$ over all $t \in \mathbf{t}_p$ (virtual waiting times) or $i_p \in \mathbf{I}_p$ (observed waiting times). The MAEs in the experiment were evaluated for different values of τ.

3.4.3 Accuracy

Table 3.4 contains the MAE and CPU times of the methods under study, averaged over all instances in the test set, for different settings of τ and C^2. For the simulation-based methods (SIM-OWM and SIM-VIRT), we control the accuracy by varying the number of replications R. For G-RAND, we vary the granularity parameter Δ_g. This section presents the results for the most extreme settings of R (i.e., 250 and 8000) and Δ_g (i.e., 0.125 and 2). The accuracy results for the other values of R and Δ_g are discussed in further detail when we study the trade-off with computational cost in Section 3.4.4.

Figure 3.2 and Table 3.5 provide insights into the differences in MAE across the methods, for $\tau = 10$ minutes (other settings for τ yield similar results). Table 3.5 shows that overall, G-RAND provides the highest accuracy (for $\Delta_g = 0.125$), followed by SIM-VIRT and SIM-OWM (for $R = 8000$). As expected, accuracy deteriorates as R decreases or Δ_g increases, as Table 3.5 indicates. However, MOL performs poorly, largely because it is unable

Value for C^2		$\tau = 0$			$\tau = 10$ min			$\tau = 20$ min			CPU time (sec)		
		0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2
SIM-VIRT	$R = 250$	0.021	0.021	0.021	0.015	0.014	0.014	0.011	0.010	0.011	0.7254	0.6638	0.7736
	$R = 500$	0.015	0.015	0.015	0.011	0.010	0.011	0.008	0.007	0.008	1.4852	1.3017	1.5078
	$R = 1000$	0.011	0.010	0.011	0.008	0.007	0.007	0.005	0.006	0.006	3.0859	2.5385	2.9744
	$R = 2000$	0.008	0.008	0.008	0.005	0.005	0.005	0.004	0.004	0.004	5.8304	5.0037	5.8685
	$R = 4000$	0.005	0.005	0.005	0.004	0.004	0.004	0.003	0.003	0.003	11.793	10.034	11.689
	$R = 8000$	0.004	0.004	0.004	0.003	0.003	0.003	0.002	0.002	0.002	23.267	20.013	23.626
SIM-OWM	$R = 250$	0.029	0.028	0.029	0.022	0.020	0.019	0.016	0.014	0.014	2.0141	1.8856	2.0295
	$R = 500$	0.020	0.020	0.021	0.015	0.014	0.014	0.011	0.010	0.010	3.7657	3.5893	3.9437
	$R = 1000$	0.015	0.015	0.015	0.011	0.010	0.010	0.008	0.007	0.007	7.3872	7.2263	7.7692
	$R = 2000$	0.011	0.011	0.011	0.008	0.007	0.007	0.006	0.005	0.005	14.674	14.357	15.329
	$R = 4000$	0.008	0.008	0.008	0.006	0.005	0.005	0.004	0.004	0.003	29.011	27.491	28.944
	$R = 8000$	0.006	0.006	0.005	0.004	0.004	0.003	0.003	0.003	0.002	59.243	53.730	54.319
MOL		0.014	0.010	0.011	0.015	0.010	0.009	0.012	0.008	0.007	111.67	81.904	98.078
G-RAND	$\Delta = 0.125$	0.010	0.002	0.002	0.012	0.001	0.001	0.009	0.000	0.001	2331.0	13.974	2791.1
	$\Delta = 0.250$	0.012	0.004	0.004	0.014	0.001	0.002	0.010	0.000	0.001	1180.9	7.3361	1421.3
	$\Delta = 0.500$	0.016	0.007	0.007	0.017	0.002	0.002	0.012	0.001	0.001	530.26	3.6450	740.51
	$\Delta = 1.000$	0.025	0.014	0.013	0.024	0.004	0.003	0.015	0.002	0.002	218.35	1.4063	259.93
	$\Delta = 2.000$	0.045	0.026	0.023	0.041	0.007	0.006	0.022	0.003	0.003	113.47	0.8978	137.16

Table 3.4: MAE and CPU time (sec), as a function of C^2

to predict performance accurately at moments in time that follow capacity changes. As an illustration, Figure 3.3 plots the probability $\Pr(W_t > 10)$ over the time horizon, for a given problem instance. The sudden surges and decreases in the probability of excessive waiting coincide with capacity changes.

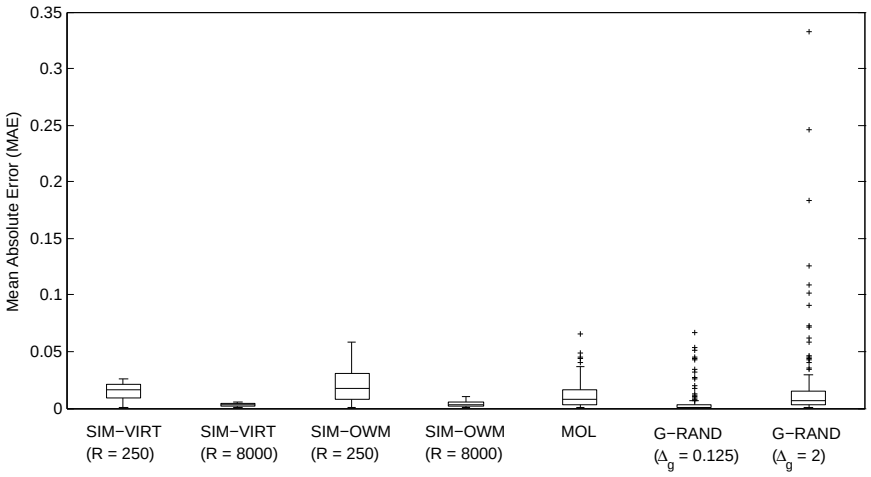


Figure 3.2: MAE, averaged over all Instances and C^2 for $\tau = 10$.

	SIM-VIRT $R = 250$	SIM-VIRT $R = 8000$	SIM-OWM $R = 250$	SIM-OWM $R = 8000$	MOL	G-RAND $\Delta_g = 0.125$	G-RAND $\Delta_g = 2$
Min	0.00032	0.00007	0.00015	0.00003	0.00003	0.00002	0.00012
Avg	0.01461	0.00265	0.02052	0.00375	0.01137	0.00469	0.01789
Med	0.01581	0.00294	0.01758	0.00324	0.00815	0.00093	0.00722
Max	0.02629	0.00551	0.05872	0.01048	0.06515	0.06667	0.33246

Table 3.5: MAE, averaged over all instances and C^2 for $\tau = 10$.

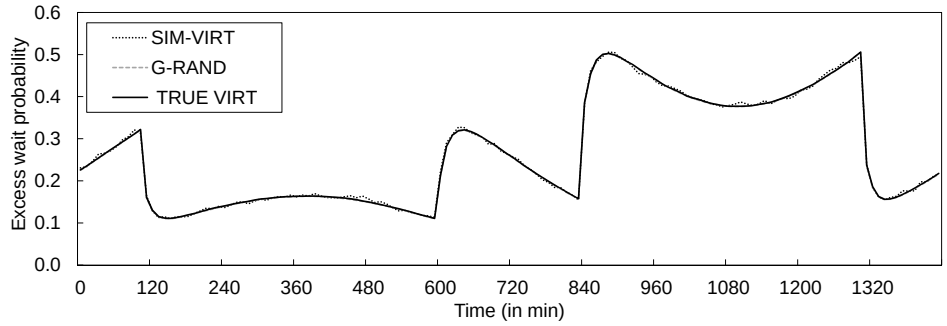
Figure 3.3(b) illustrates the shortfalls of MOL. The performance curve obtained by MOL lags the curves obtained by the other methods; this effect can be explained by the stationary approximation being applied. A key insight is that a capacity increase or decrease at any time t affects performance already before the change takes place – namely, from time $(t - \tau)$ onward.

Therefore, the impact on the probability of excessive waiting occurs earlier as τ increases. Clearly MOL ignores this effect, because only the capacity during an interval i_p gets taken into consideration when evaluating the stationary model. Accordingly, MOL disregards any subsequent capacity changes that affect performance in the interval. This effect only occurs for $\tau > 0$ and becomes more pronounced as τ grows large, as further illustrated in Figure 3.4, which shows the graphs for $\tau = 0$ and $\tau = 20$.

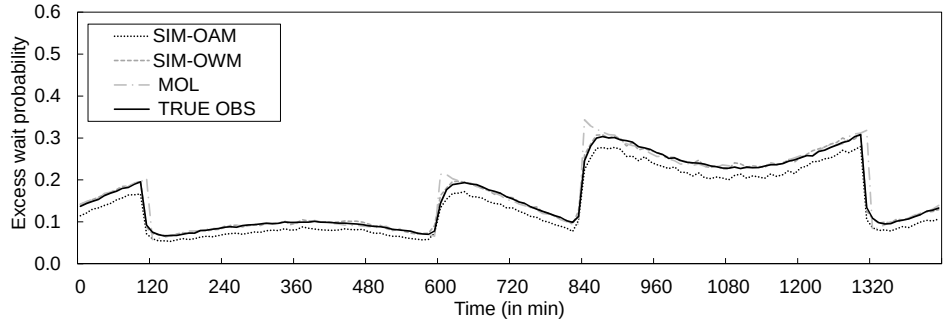
Figure 3.4 hints on an additional shortfall of the MOL approximation. In Figure 3.4(a), we observe that MOL consistently underestimates the probability of excessive waiting following a capacity increase, but overestimates it following a capacity decrease. This second observation can again be explained by the stationary approximation in MOL, which assumes an immediate surge in the departure rate if capacity increases. In reality, though, the change in the departure rate is not immediate but lagged by the expected service time. Consequently, MOL underestimates the queues at these moments, resulting in an underestimation of the probability of excessive waiting. The overestimation of $\Pr(W_t > 0)$ following a capacity decrease follows from MOL's inability to mimic the exhaustive service policy (i.e., servers do not work overtime; the effective utilization is higher). The results thus indicate that MOL should be used with caution, especially when the capacity fluctuations are frequent/substantial and the system is heavily loaded.

Moreover, Figure 3.3(b) and Figure 3.4 reveal the difference between SIM-OAM and SIM-OWM. The probability of excessive waiting for SIM-OWM consistently exceeds the curve of SIM-OAM in our experiments (this consistent with the findings of Maman [173]). As such, SIM-OWM results in more stringent performance constraints. A formal description of the relation between SIM-OAM and SIM-OWM is given in Appendix B (we also provide an example where SIM-OAM may exceed SIM-OWM in some cases).

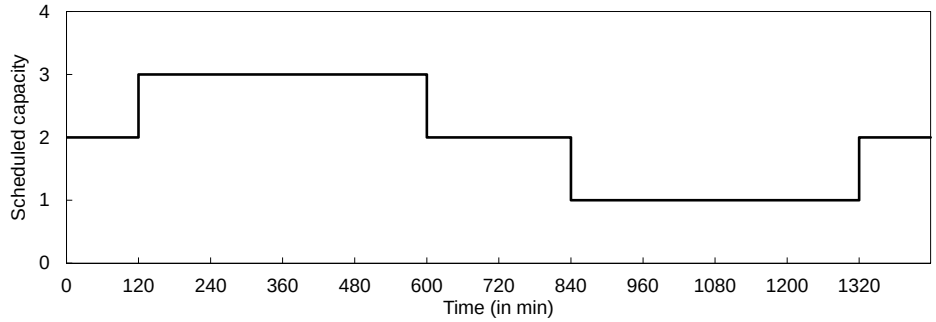
From the results in Table 3.4, it follows that the squared coefficient of variation in the service and abandonment processes is not a significant determinant of accuracy, except for G-RAND for which accuracy deteriorates when $C^2 < 1$. This observation is not surprising: as Creemers et al. [57] note, low values of C^2 require lower values of Δ_g to maintain accuracy.



(a) Virtual waiting times

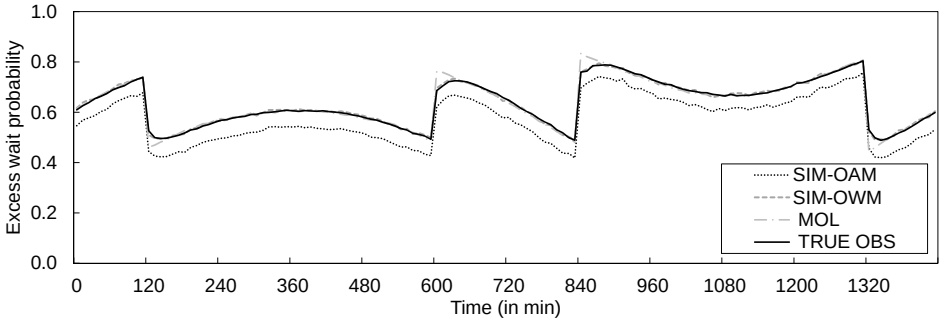


(b) Observed waiting times

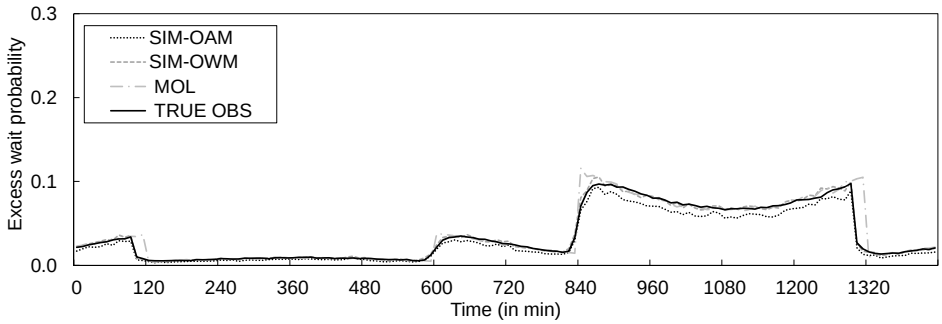


(c) Scheduled capacity

Figure 3.3: $\Pr(W_t > 10)$ for a given problem instance.
Problem instance: $\mu = 6$, $\theta = 3$, $\bar{s} = 2$, $\bar{\rho} = 0.95$, and $C^2 = 1$ ($R = 8000$, $\Delta_g = 0.125$).



(a) Observed waiting times $\tau = 0$



(b) Observed waiting times $\tau = 20$

Figure 3.4: $\Pr(W_t > \tau)$ for a given problem instance.
Problem instance: $\mu = 6$, $\theta = 3$, $\bar{s} = 2$, $\bar{\rho} = 0.95$, and $C^2 = 1$ ($R = 8000$, $\Delta_g = 0.125$).

3.4.4 Computational cost and trade-off with accuracy

Figure 3.5 and Table 3.6 offer an indication of the CPU time required to obtain the probability of excessive waiting. Computation times are independent of the threshold value τ, because we used a single experiment to obtain the probabilities of excessive waiting for different thresholds simultaneously. The boxplots contain only the results for the most extreme values of R (i.e., 250 and 8000) and Δ_g (i.e., 0.125 and 2). On the whole, SIM-VIRT is the fastest method, whereas G-RAND demands the greatest computational effort (especially when Δ_g is small). The computation times for MOL are disappointing, though they likely could be improved were we to use dedicated approximations for the $M/G/s + G$ queue (instead of simulation). Using approximations, however, affects the accuracy of the method.

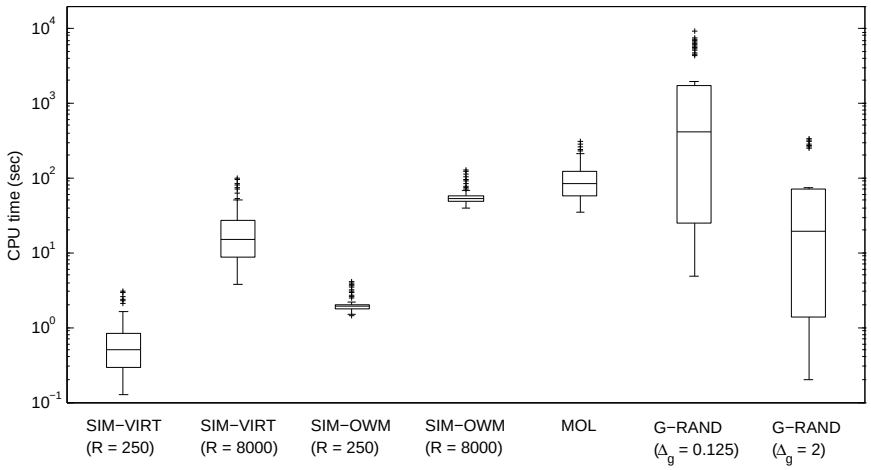


Figure 3.5: CPU Time (sec) for all τ, averaged over all values of C^2.

Figure 3.6 depicts the trade-off between MAE and CPU time, in the simulation-based methods as well as in G-RAND (MOL yields a single observation point). In the simulation-based methods, computation times and accuracy are influenced by the number of replications R, while in G-RAND, they are affected by the granularity parameter Δ_g. We observe that G-RAND outperforms SIM-VIRT only when $C^2 = 1$. For $C^2 > 1$, G-RAND

	SIM-VIRT $R = 250$	SIM-VIRT $R = 8000$	SIM-OWM $R = 250$	SIM-OWM $R = 8000$	MOL	G-RAND $\Delta_g = 0.125$	G-RAND $\Delta_g = 2$
Min	0.125	3.713	1.451	40.217	35.061	4.780	0.200
Avg	0.721	22.302	1.976	55.764	97.219	1712.032	83.842
Med	0.508	15.233	1.887	53.049	84.445	404.610	19.250
Max	3.088	100.730	4.087	125.290	302.680	9024.200	341.670

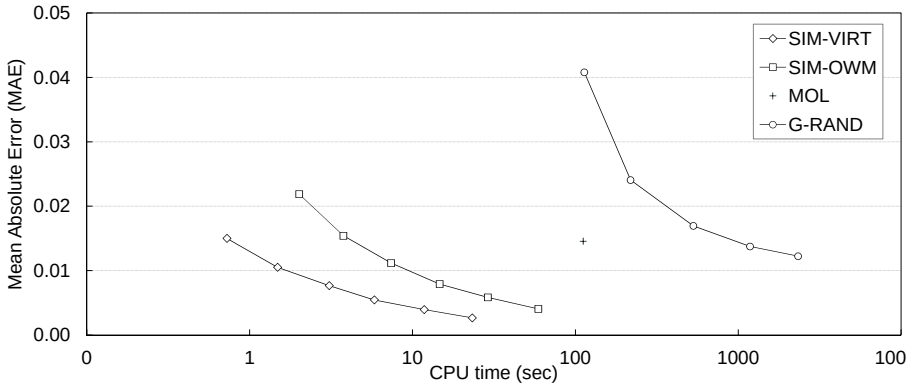
Table 3.6: CPU Time (sec) for all τ, averaged over all values of C^2

.

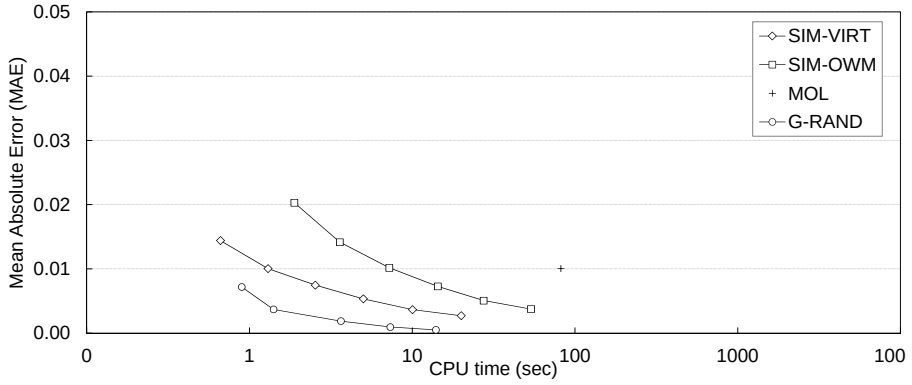
obtains a higher average accuracy than SIM-VIRT, though at a markedly higher computational cost. In that case, G-RAND is preferable only if accuracy is more important than computation time. For $C^2 < 1$, the trade-off curve of G-RAND is unfavorable (low accuracy at high computational cost). As illustrated by Figure 3.7, the difference in performance grows larger as τ increases. Accuracy might be improved by reducing Δ_g, but doing so would increase CPU time. In summary, G-RAND only outperforms SIM-VIRT for $C^2 = 1$. For other values of C^2, SIM-VIRT is the better choice, yielding low MAE at low computational cost.

3.5 Conclusion

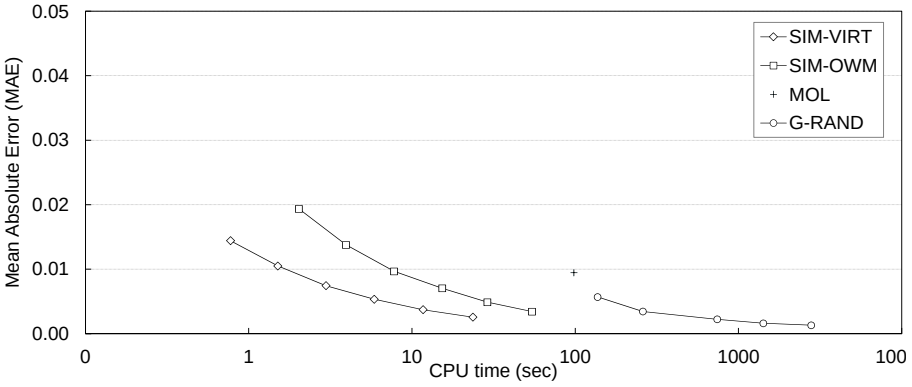
In this chapter, we explore different methods to evaluate the time-dependent probability of excessive waiting in an $M_t/G/s_t + G$ queueing system with an exhaustive service policy. We evaluate four methods: (1) simulation based on virtual waiting times (SIM-VIRT), (2) simulation based on observed waiting times (SIM-OWM), (3) the modified offered load approximation (MOL), and (4) the randomization approach suggested by Creemers et al. [57] (G-RAND). The results show that SIM-VIRT consistently yields accurate results with limited computation time, such that it outperforms the other methods, except when $C^2 = 1$, in which case G-RAND provides a more favorable time-accuracy trade-off. Moreover, the MOL approximation performs rather poorly and displays systematic error each time the capacity changes, so it should be used with extreme caution (especially when capacity changes frequently or the maximum allowed waiting time is greater).



(a) $C^2 = 0.5$

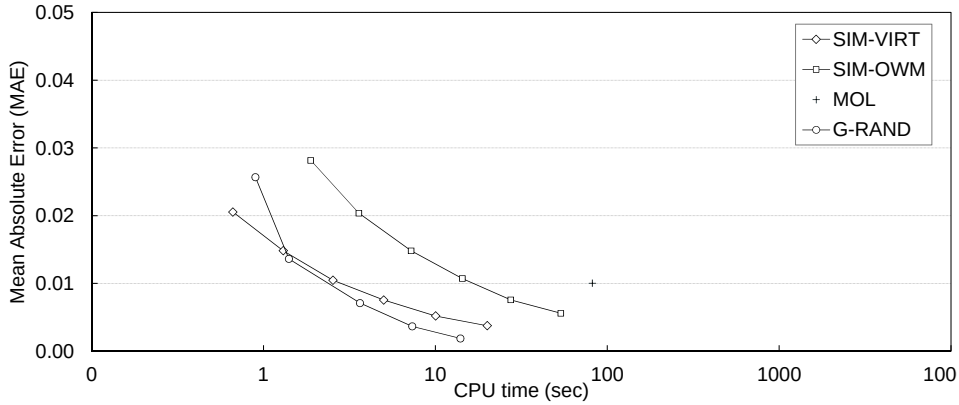


(b) $C^2 = 1$

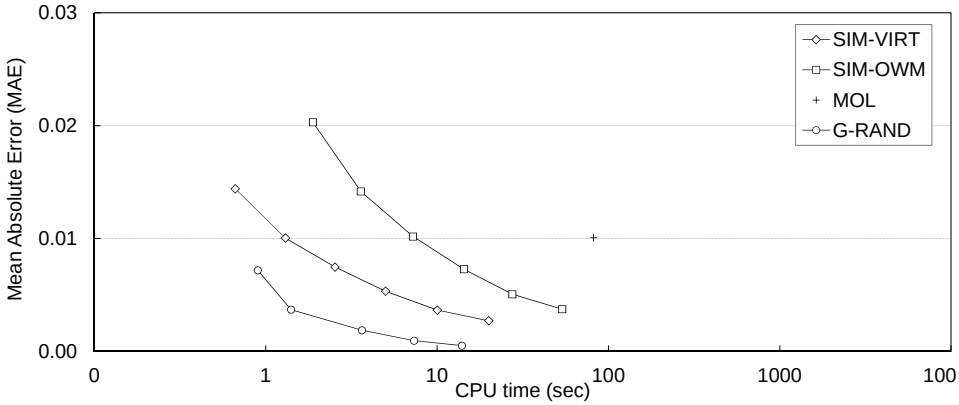


(c) $C^2 = 2$

Figure 3.6: Trade-off between accuracy and computation time (for $\tau = 10$).



(a) $\tau = 0$



(b) $\tau = 20$

Figure 3.7: Trade-off between accuracy and computation time, averaged over all instances for $C^2 = 1$.

Chapter 4

Controlling excessive waiting times in small service systems with time-varying demand: an extension of the ISA algorithm

This chapter presents a simulation-based staffing method that enables to stabilize the *probability of excessive waiting* (i.e., the probability that the waiting time exceeds a maximum acceptable value) throughout the day, in a single-stage multiserver system with customer abandonments. The suggested method is inspired by the Iterative Staffing Algorithm (ISA), proposed by Feldman et al. [78], which focuses on stabilizing the *delay probability* throughout the day (note that this corresponds to a maximum acceptable waiting time of zero). The use of discrete-event simulation provides distinct advantages over analytical methods, such as increased flexibility in modeling assumptions and the ability to control the precision of the results. The downside is that evaluation through simulation tends to be more time-consuming. Using the simulation methods described in Chapter 3, we can efficiently evaluate the probability of excessive waiting. The staffing update function of

the original ISA algorithm is adjusted to account for the relatively small system scale that characterizes, for example, emergency departments, retail stores or small call centers.

Our experiments indicate that our method (which we call ISA(τ)) succeeds in finding a staffing vector that meets the performance constraint, irrespective of system size. Large-scale systems (for which the number of servers required is in the order of 100) and extremely small-scale systems (requiring only 1-2 servers) can be solved, although the computation time increases with the problem size. A solution can be obtained for exponential as well as general service and abandonment time distributions, and staffing intervals are taken into consideration. We check the method for several service policies (exhaustive and preemptive) and evaluate different waiting time thresholds.

The remainder of this chapter is organized as follows: we briefly discuss the related literature in Section 4.1, a detailed description of ISA(τ) then follows in Section 4.2. We present computational results of ISA(τ) in Section 4.3 and compare the approach to stationary approximation techniques available in the literature. Section 4.4 summarizes our results, along with some directions for future research.

4.1 Related literature

In this section, we briefly describe the literature that is most relevant to our research. As is detailed in Chapter 2, systems with time-dependent arrival rates are commonly approximated by one (or more) related stationary model(s). In Section 4.3, the performance of ISA(τ) is tested against a number of these approaches, so we briefly discuss the stationary approximations that are most common in our context in Section 4.1.1. Section 4.1.2 then proceeds with a detailed description of the iterative staffing algorithm proposed in Feldman et al. [78], that can be considered as the starting point of the ISA(τ) approach.

4.1.1 Stationary approximations

In the *Pointwise Stationary Approximation* (PSA), the arrival rate at each time instant is plugged into a separate stationary model to obtain steady state performance measures for each moment in time [91, 131, 246]. PSA is most appropriate in large-scale systems with limited nonstationarity in the arrival process, characterized by high service rates, high targeted quality of service, and low to moderate loads. To improve performance in case of low service rates, a *lagged* variant of PSA (denoted LagPSA) has been proposed [92, 96]. This approach is identical to PSA, except for the use of a lagged arrival rate $\lambda(t - E[S])$, where $E[S]$ (i.e., the expected service time) represents a *time lag* (further theoretical background supporting this choice, can be found in Eick et al. [70]).

In the *Modified Offered Load* (MOL, [130, 184, 185]) approach, a stationary model is solved at each point in time, using a modified arrival rate that equals the product of the service rate and the infinite server offered load (i.e., the number of servers that would be used if infinitely many servers were available). As the number of servers decreases, the MOL approximation becomes less accurate because of a lower resemblance to the infinite server system [185].

The PSA, lagged PSA and MOL approaches vary staffing levels continuously and do not account for the presence of staffing intervals. To this purpose, two refinements to the PSA approach have been proposed: segmented PSA and the *Stationary Independent Period-by-Period* (SIPP) approaches. In the *Segmented PSA* approximation, the staffing levels are set equal to the maximum of the PSA staffing requirements over the staffing interval [99]. The SIPP approach [96, 98] uses a stationary model in each staffing interval, with the arrival rate averaged over that interval. As shown in Green et al. [96], SIPP does not perform well when staffing intervals are long or when the arrival rate changes substantially over the staffing interval.

Further refinements to the SIPP approach have been proposed in Green et al. [96] and Green et al. [97]: Lag SIPP (which uses a lagged arrival rate), SIPPmax (which uses the maximum arrival rate over the staffing interval instead of the average, and hence coincides with Segmented PSA),

and lagged SIPPmax (which is a combination of both). Lagged SIPP and lagged SIPPmax tend to have better performance than SIPP and SIPPmax [97].

Once the nonstationary system has been transformed into one or more stationary models, approximations are often needed to obtain the steady state performance measures, in particular when service times and abandonment times are generally distributed (as in our setting). For further details on approximations for the $M/G/s+G$ model, we refer to Iravani and Balcioğlu [124] and Whitt [249]. Often however, explicit performance calculations are avoided by using a rule of thumb, known as the *square-root staffing rule* (SRS; see [85, 247, 152]). The main benefit of SRS lies in its simplicity and robustness: at each time instant, the staffing level (denoted s) is determined as the offered load (denoted m) augmented with an amount of safety capacity (cf. Expression 4.1). The required safety capacity is proportional to the offered load, and depends on the desired quality of service (which is reflected in the quality of service parameter β):

$$s = m + \beta\sqrt{m} \tag{4.1}$$

The appropriate β can be obtained through the inverse of the Halfin-Whitt delay function (for $M/M/s$ models, cf. Halfin and Whitt [106]) or the Garnett delay function (for $M/M/s+M$ models, cf. Garnett et al. [88]). An extension towards $M/M/s+G$ models can be found in Zeltyn and Mandelbaum [257]. As a rule of thumb, the SRS rule is easy to apply. However, it provides no firm guarantee that the desired performance constraint is actually met.

A key disadvantage of most stationary approximations is that the use of stationary models implicitly assumes that (1) delays between separate intervals are statistically independent, (2) steady state is reached in each interval, (3) the arrival rate remains constant over the staffing interval, and (4) no overloading is present within any given interval, as this would cause instability in the stationary model (unless abandonments occur) [96]. These assumptions are not always valid.

4.1.2 The Iterative Staffing Algorithm (ISA)

In Feldman et al. [78], a promising simulation-based technique for determining staffing requirements in time-varying queues is proposed. As the name suggests, the Iterative Staffing Algorithm (ISA) repeatedly evaluates and alters the staffing function, until the desired performance is attained. For each staffing function, system performance is evaluated by means of simulation and the staffing level is updated based on the observed performance. This sequence of evaluating performance and updating staffing levels is called an iteration.

Performance is expressed in terms of a constraint on the delay probability, that is, the delay probability must lie below a target value α at all time instants:

$$\Pr(W_t > 0) \leq \alpha \qquad 0 \leq t \leq T\,. \tag{4.2}$$

Equivalently, the delay probability equals the probability that the number of customers in the system at time t, N_t, is larger than or equal to the available capacity at that time, leading to the following constraint:

$$\Pr(N_t \geq s_{i_s}) \leq \alpha \qquad \forall\ t \in \mathbf{t}_s\,, \tag{4.3}$$

where $t \in \mathbf{t}_s$ represents the start time of staffing interval i_s. The ISA assumes staffing changes can be made almost continuously. The planning horizon T is divided into very small intervals: staffing changes are only allowed at the start of each interval and the number in system is evaluated once every staffing interval. ISA then proceeds as follows. Initially, all staffing levels are set equal to an arbitrarily large number. Subsequently, system performance is simulated by performing a fixed number of independent replications, which results in a distribution of the number of customers in system at each moment in time. Then, staffing levels are improved (simultaneously for all staffing intervals) such that in each staffing interval $i_s \in \mathbf{I}_s$, the staffing level corresponds to the smallest value of s_{i_s} satisfying the performance requirement in Expression 4.3. Formally, the evaluation of the distribution of the number of patients in system at the start of staffing interval i_s in iteration k (denoted $N_{t,k}$ with $t \in \mathbf{t}_s$) determines the staffing

level in interval i_s in iteration $k+1$ (denoted $s_{i_s,k+1}$):

$$s_{i_s,k+1} = \arg\min\{j \in \mathbb{N} : \Pr(N_{t,k+1} \geq j) \leq \alpha\} \qquad \forall\ i_s \in \mathbf{I}_s\,, \tag{4.4}$$

where $t \in \mathbf{t}_s$ represents the start time of staffing interval i_s. The algorithm stops when the staffing changes in subsequent iterations become sufficiently small for all staffing intervals (i.e., staffing levels differ by at most 1, for all i_s).

The major advantage of the ISA lies in the use of simulation to evaluate system performance. As a result, the appropriateness of the staffing function generated by ISA is validated automatically (that is, under the assumption that the simulation model is adequate). Moreover, the method has potential to be applied to general settings, for which analytical results are no longer available. However, some aspects make the traditional ISA less appropriate in small-scale contexts:

- As discussed in Feldman et al. [78], the delay probabilities obtained by the original ISA tend to be less stable in periods with low demand, as even a small change in capacity has a substantial impact on performance. Moreover, the conventional stopping rule of the ISA might pose problems: the algorithm stops when the change in staffing requirements is at most 1 in all staffing intervals and thus, staffing changes of +/- 1 server are disregarded. In small-scale systems however, the addition or removal of one server can result in substantial differences in performance.

- Secondly, the ISA does not explicitly deal with the length of the staffing intervals, i.e., the time period over which capacity remains constant (all examples used in Feldman et al. [78] assume small staffing intervals with a length of $0.1/\mu$). As the number of customers in system is measured only once every interval, it can be expected that an increase in the staffing interval length will lead to a decrease in accuracy, which will negatively impact the algorithm's performance. In the method we present in Section 4.2, this problem is addressed by making a distinction between staffing intervals and (smaller) performance intervals.

- Finally, the results in Feldman et al. [78] indicated that a staffing function that stabilizes delay probability does not automatically stabilize other performance measures (such as abandonment probabilities, average queue lengths and average waiting times). Focusing on the probability of excessive waiting is more relevant and allows more flexibility, as the decision maker can decide both on the waiting time threshold that should be met (τ), and on the target service level (α). This, however, implies that staffing levels can no longer be set using Expression 4.4.

These observations justify the search for a method which (1) is suitable for small-scale systems, (2) is capable of dealing with staffing intervals over which the capacity remains constant and (3) emphasizes low probabilities of excessive wait rather than delay probabilities. In the next sections, a more detailed description of the ISA(τ) method is given.

4.2 ISA(τ) algorithm

4.2.1 Notation

Notations that are frequently used throughout this chapter are listed in Table 4.1. Let $s_{i_s,k}$ represent the staffing level in staffing interval $i_s \in \mathbf{I}_s$ at the start of k^{th} iteration of the algorithm, so that $\mathbf{s}_k$ denotes the staffing vector. The performance vector corresponding to $\mathbf{s}_k$ is represented by $\mathbf{P}_k$; its elements $P_{t,k}$ denote the excess wait probability for a customer arriving at time $t \in \mathbf{t}_p$. The time-average of $P_{t,k}$ is $\bar{P}_k$. For a given staffing interval i_s starting at $t \in \mathbf{t}_s$ and ending at $t+\Delta_s$, $P^{\max}_{i_s,k}$ denotes the maximum value of the excess wait probabilities $P_{i_s,k}$ that directly result from the capacity $s_{i_s,k}$. Formally, for each staffing interval $i_s \in \mathbf{I}_s$ (starting at $t \in \mathbf{t}_s$ and ending at $t+\Delta_s$), $P^{\max}_{i_s,k} = \max_{j \in \mathbf{t}_p} \{P_{j,k} : j \in \quad [t-\tau, (t+\Delta_s)-\tau[\}$. Indeed, if a capacity shock occurs at $t \in \mathbf{t}_s$, the excess wait probabilities are affected for all arrivals after $(t-\tau)$, as τ represents the time window within which these patients should start service in order to have a waiting time below the threshold τ (as illustrated in Figure 4.1). Thus, the capacity in a staffing

interval i_s (starting at $t \in \mathbf{t}_s$ and ending at $t+\Delta_s$) has a direct[1] effect on the performance of patients arriving during interval $[t-\tau, (t+\Delta_s)-\tau[$. Each vector $\mathbf{P}_k$ (containing excess wait probabilities at times $t \in \mathbf{t}_p$) can thus be translated into vector $P^{\max}_{i_s,k}$ (containing maximum excess wait probabilities per staffing interval i_s).

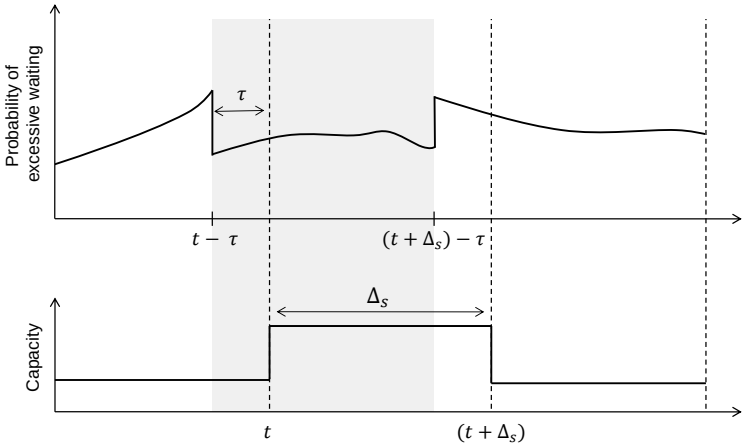


Figure 4.1: Interval over which capacity impacts performance

For a given staffing vector $\mathbf{s}_k$, e_k denotes the number of staffing intervals for which the performance target is not met (i.e., $P^{\max}_{i_s,k} > \alpha$). Additionally, we associate a staffing cost $c_{s,k}$ with each staffing vector $\mathbf{s}_k$, and define c^*_s to be the lowest cost found so far, for a solution that meets the performance constraint at all times. The cost (expressed in man-hours) for using one unit of capacity during one staffing interval is denoted as u.

4.2.2 Performance measurement through simulation

The probability of excessive waiting, $\Pr(W_t > \tau)$, is measured from the simulation model by means of *virtual waiting times*, as described in Chapter 3 (we briefly repeat it here). The virtual waiting time corresponds to the

[1]It is clear that an indirect effect is present as well; i.e., the capacity level in any staffing interval also has an impact on the performance at all later time instants, through the number of customers in system.

Notation		
t	:	Time
$\mathbf{t}_s$	:	Set of staffing interval start times
$\mathbf{t}_p$	:	Set of performance interval start times
Δ_p	:	Performance interval length
Δ_s	:	Staffing interval length (assumed to be divisible by Δ_p)
k	:	Iteration index
i_s	:	Staffing interval index
$\mathbf{I}_s$	:	Set of staffing intervals
r	:	Replication index
R	:	Total number of replications in eac simulation run
W_t	:	Waiting time at time t
$W_{t,r}$	:	Waiting time at time t, in replication r
τ	:	Waiting time threshold value
α	:	Target w.r.t. waiting time service level
N_t	:	Number of customers in the system (in queue and in service) at time t
$N_{t,k}$	:	Number of customers in the system (in queue and in service) at time t, i.e., at the start of staffing interval i_s, with $t \in \mathbf{t}_s$
N^t_{t+w}	:	Number of customers that arrived before time t that are still in system at time $t+w$
$\mathbf{s}_k$	:	Staffing vector in iteration k
$s_{i_s,k}$	:	Element of $\mathbf{s}_k$ in iteration k, with staffing interval $i_s \in \mathbf{I}_s$.
$\mathbf{s}^{\mathsf{init}}$	:	Initial staffing vector
$A_{i_s,k}$	:	Amplification factor in staffing interval i_s, in iteration k
$\mathbf{P}_k$	:	Performance vector corresponding to $\mathbf{s}_k$
$P_{t,k}$	:	Elements of $\mathbf{P}_k$, with $t \in \mathbf{t}_p$
$\bar{P}_k$	:	Time-average of $P_{t,k}$
$P^{\max}_{i_s,k}$	:	maximum value of the excess wait probabilities $P_{i_s,k}$ within interval $[t-\tau, (t+\Delta_s)-\tau[$, with $t \in \mathbf{t}_s$
$c_{s,k}$	:	Staffing cost in iteration k
c^*_s	:	Lowest staffing cost found so far
u	:	Cost (expressed in man-hours) for using one unit of capacity during one staffing interval
e_k	:	Number of staffing intervals with $P^{\max}_{i_s,k} > \alpha$, for a given staffing vector $\mathbf{s}_k$

Table 4.1: Chapter 4: notations

time between t and the earliest time at which a (scheduled) server becomes available, because all customers that arrived before t have been served [102, 159, 174]:

$$W_t = \min\{w : \left(N^t_{t+w} \leq s_{t+w} - 1\right) \wedge (w \geq 0)\},$$

with s_{t+w} the capacity at time $t + w$ and N^t_{t+w} the number of customers arrived before time t that are still in system at time $t + w$. Note that the virtual waiting time is measured at a particular time instant (as opposed to *observed* waits, which are measured over an interval). The virtual waiting time distribution can be measured in a straightforward way through simulation. We insert a virtual (dummy) customer into the system at each time $t \in \mathbf{t}_p$ in replication r, such that the virtual waiting time $W_{t,r}$ equals the time at which this dummy customer would enter service. Let R represent the total number of replications in the simulation run. Define $\delta_{t,r}$ as a binary variable that signals whether the virtual waiting time exceeds the target τ for a given time t and replication r:

$$\delta_{t,r} = \begin{cases} 1 & \text{if } W_{t,r} > \tau, \\ 0 & \text{otherwise } . \end{cases}$$

The probability of excessive waiting at time t then can be estimated as:

$$\Pr(W_t > \tau) = \frac{1}{R} \sum_{r=1}^{R} \delta_{t,r} \, .$$

4.2.3 Optimization procedure

The ISA(τ) algorithm starts with an initial staffing solution $\mathbf{s}^{\text{init}}$ that is not necessarily feasible. In our experiments, the capacity in each staffing interval is set equal to the staffing solution obtained by applying the SRS rule[2] to each interval, using the lagged SIPP arrival rate and β that results from the Garnett Delay function for $M/M/s + M$ systems [88].

A first phase in the algorithm (PHASE I or "exploration phase", summarized in Algorithm 4.1) aims at quickly finding a staffing vector for which

[2]We also obtained good results if the capacity in each staffing interval is set equal to $\lceil \bar{\lambda}/\mu \rceil$, with $\bar{\lambda}$ the average arrival rate, especially for small-scale systems.

performance is *close* to the target (but not necessarily below the target value at all times). To this purpose, the current staffing level function $s_{i_s,k}$ is altered iteratively, based on the simulation output.

During each iteration k of the algorithm, $s_{i_s,k}$ is updated as follows:

$$s_{i_s,k+1} = \begin{cases} \lceil s_{i_s,k} A_{i_s,k} \rceil & \text{if } A_{i_s,k} \geq 1, \quad \forall\ i_s \in \mathbf{I}_s \\ \lfloor s_{i_s,k} A_{i_s,k} \rfloor & \text{if } A_{i_s,k} < 1, \quad \forall\ i_s \in \mathbf{I}_s \end{cases} \tag{4.5}$$

where $A_{i_s,k}$ refers to an amplification factor, which is determined based on the deviation between $P^{\max}_{i_s,k}$ and the target α (in percent):

$$A_{i_s,k} = 1 + \frac{P^{\max}_{i_s,k} - \alpha}{\alpha k} \qquad \forall\ i_s \in \mathbf{I}_s\,, \tag{4.6}$$

with $P^{\max}_{i_s,k}$ derived from the simulation results, as described in Section 4.2.2.

Values $P^{\max}_{i_s,k}$ (above) target will result in an $A_{i_s,k}$ below (above) 1 and thus a decrease (increase) in capacity in the corresponding interval (note that due to the rounding in Expression 4.5, capacity is always increased or decreased with at least one unit).

The use of the scaling factor k in the denominator of Expression 4.6 ensures that $A_{i_s,k}$ approaches 1 (for all i_s) as the number of iterations increases. This forces the algorithm to decrease the size of the staffing changes as it progresses, eventually switching to unit-size changes and converging to a final staffing vector (despite the fact that possible deviations from the target may still remain). In fact, the choice of the scaling factor in the denominator of Expression 4.6 is rather arbitrary; other factors may be considered, such as $k/2$ or k^2. Especially in large-scale systems, high values for the factor should be used with caution: they may lead the algorithm to switch to unit-size capacity changes too soon. This issue is illustrated numerically in Section 4.3.1.

We allow the algorithm to stop exploring when cycling occurs (meaning that the staffing vector put forward in the current iteration has already been assessed during a previous iteration)[3]. This stop criterion usually yields

[3]Recall that the original ISA uses a different terminating condition: it stops when the staffing level changes at most with one unit in each interval, compared to the previous iteration. We opt not to use this stop criterion due to the focus on small system sizes, where one unit capacity changes may occur more frequently (causing the algorithm to stop prematurely).

Algorithm 4.1 ISA(τ): PHASE I : EXPLORATION

Initial staffing vector: $\mathbf{s}_0 = \mathbf{s}^{\text{init}}$

Initialize stop criterion: *stop* $\leftarrow$ FALSE

Initialize iteration counter: $k \leftarrow 0$

while *stop* = FALSE **do**

$k \leftarrow k + 1$

Simulate staffing vector $\mathbf{s}_k$ (result: performance vector $P^{\max}_{i_s,k}$)

Update capacity in all $i_s \in \mathbf{I}_s$:

$$A_{i_s,k} \leftarrow 1 + \frac{P^{\max}_{i_s,k} - \alpha}{\alpha k}$$

$$s_{i_s,k+1} \leftarrow \begin{cases} \lceil s_{i_s,k} A_{i_s,k}) \rceil & \text{if } A_{i_s,k} \geq 1, \quad \forall\, i_s \in \mathbf{I}_s \\ \lfloor s_{i_s,k} A_{i_s,k} \rfloor & \text{if } A_{i_s,k} < 1, \quad \forall\, i_s \in \mathbf{I}_s \end{cases}$$

Determine $\bar{P}_k$ and MA_k

if $\exists j < (k+1) | \forall\, i_s : s_{i_s,j} = s_{i_s,k+1}$ **then**

stop $\leftarrow$ TRUE; repetition in staffing levels, so proceed to PHASE II

else if $\forall\, j = k, k-1, \ldots, k-4 : \bar{P}_j \in [\text{MA}_k - 0.025, \text{MA}_k + 0.025]$ **then**

stop $\leftarrow$ TRUE; Performance is stabilizing, so proceed to PHASE II

good results for small-scale systems.

For large-scale systems, we propose to use an additional stop criterion. We observed that in these systems, many iterations may be needed before cycling in the staffing levels occurs, while the excess wait probability usually stabilizes far more quickly. We thus suggest to keep track of the average probability of excessive waiting over the time horizon in each iteration ($\bar{P}_k$) and stop the algorithm if the most recent values of $\bar{P}_k$ consistently are close to the moving average of $\bar{P}_k$. Formally, the algorithm terminates if $\forall\, j = k, k-1, ..., k-4 : \bar{P}_j \in [\text{MA}_k - 0.025, \text{MA}_k + 0.025]$, with MA_k the moving average of $\bar{P}_k$ over the past 10 iterations. An illustration of the typical evolution of MA_k and $\bar{P}_k$ throughout the algorithm in a large-scale problem setting is given in Figure 4.2. The ranges around MA_7 and MA_{13} are plotted; the algorithm terminates after iteration 13 because $\bar{P}_9$ to $\bar{P}_{13}$ do not deviate more than 0.025 from MA_{13} (this criterion is not yet met in the previous iterations, e.g., in iteration 7). For large-scale systems, this additional stop criterion can substantially lower the computational time in PHASE I.

Note that the exploration phase does not necessarily result in a feasible staffing vector. Consequently, an additional fine-tuning procedure (PHASE

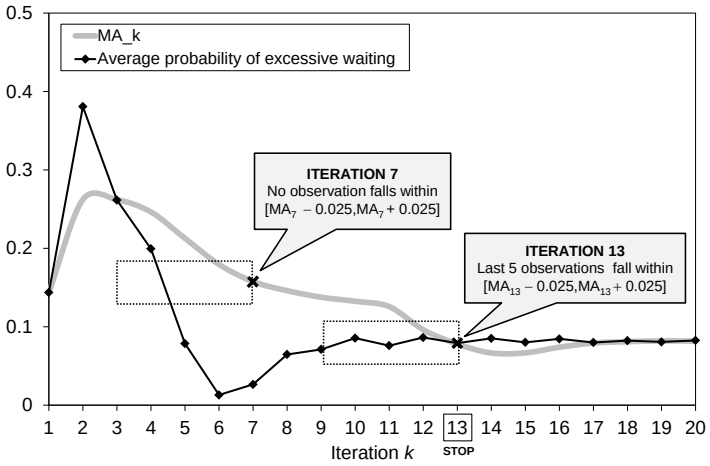


Figure 4.2: Additional stop criterion

II or "exploitation phase", see Algorithm 4.2) is needed. Phase II derives feasible solutions from the infeasible staffing levels obtained in Phase I, in hope of finding a solution which outperforms the best feasible solution found so far (if any) in terms of labor cost. To that end, all infeasible solutions encountered during Phase I are first sorted based on increasing maximum excess wait probability over the time horizon (i.e., increasing values of $\max_{i_s} \left\{P_{i_s,j}^{\max}\right\}$, where index j denotes an infeasible solution). A lower value indicates smaller deviations from the target, which makes the solution more promising to fine-tune. In case of a tie, the number of intervals with a performance constraint violation is examined. If the target is exceeded in just a limited number of intervals, a small number of capacity increases may be sufficient to obtain a feasible staffing level vector, which is appealing from a labor cost perspective.

We calculate the staffing cost that results when adding one unit of capacity to each staffing interval that causes the excess wait probability to surpass the target (we call this the projected staffing cost). Thus, in case of a tie in the maximum excess wait probability, infeasible solutions are sorted based on increasing projected staffing cost.

Next, all infeasible solutions are improved one by one (in sorted order). For a given infeasible solution j, an improved staffing vector $\mathbf{s}'_j$ (with

corresponding cost $c'_{s,j}$) is constructed by adding one unit of capacity in all staffing intervals where performance is unsatisfactory. The new staffing vector's performance is evaluated through simulation, provided that its cost is strictly lower than c^*_s (further exploiting the current infeasible solution is futile if $c'_{s,j} \geq c^*_s$). Based on the simulation results, two cases can be distinguished:

- The performance constraint is not yet met, in which case the exploitation continues. Vector $\mathbf{s}'_j$ is improved further (unit size capacity increases are made) and its performance is simulated (if the cost is lower than c^*_s).
- The performance constraint is satisfied at all times; in this case a new feasible solution is found, which is stored if it is less costly than the current best feasible solution in terms of labor cost. The exploitation of solution j is terminated and the algorithm then proceeds to the next infeasible solution in the sorted list.

Note that the procedures described in PHASE I and PHASE II are suitable for small-scale as well as large-scale systems, largely avoid cyclic behavior and moreover guarantee that the algorithm yields a staffing vector meeting the performance constraint.

4.3 Computational results

We tested the algorithm on two settings: a large-scale example, taken from Feldman et al. [78], and a small-scale setting based on real-life arrival data at the ED of a Belgian hospital. Unfortunately, no detailed process data were available from the hospital, hence we assume service times that are exponentially distributed with mean 30 minutes (similar to Green et al. (2006), where service times were chosen based on physician workload estimations that are available in Graff et al. (1993)). Table 4.2 summarizes the main characteristics for both examples; in Figure 4.3 the corresponding arrival rates are plotted. For both examples, a 24-hour time horizon is considered and performance was calculated quasi-continuously ($\Delta_p = 1$ minute). The length of the staffing interval equals 15 minutes and the waiting time thresh-

Algorithm 4.2 ISA(τ): PHASE II: EXPLOITATION

Define $c_{s,k}$ the cost of a staffing vector $\mathbf{s}_k$
Define c_s^* the cost of the cheapest feasible solution found so far
Define e_k the number of staffing intervals with $P_{i_s,k}^{\max} > \alpha$, for a given staffing vector $\mathbf{s}_k$
Define u the cost associated to using one unit of capacity during one staffing interval (expressed in man-hours)
Initialize c_s^* = cost of best feasible solution found during PHASE I (if any), ∞ otherwise

Sort all infeasible solutions j considered in PHASE I, based on
1) increasing $\max_{i_s} \{P_{i_s,k}^{\max}\}$
2) $c_{s,k} + ue_k$

for all infeasible solutions j (in sorted order) **do**

$$s'_{i_s,j} \leftarrow \begin{cases} s_{i_s,j} + 1 & \text{if } P_{i_s,j}^{\max} > \alpha \quad \forall\, i_s \in \mathbf{I}_s \\ s_{i_s,j} & \text{if } P_{i_s,j}^{\max} \leq \alpha \quad \forall\, i_s \in \mathbf{I}_s \end{cases}$$

Calculate cost of $\mathbf{s}'_j$: $c'_{s,j} \leftarrow c_{s,j} + ue_j$

while $c'_{s,j} < c_s^*$ **do**
Simulate $\mathbf{s}'_j$

if $\max_{i_s} \{P_{i_s,j}^{\max}\} > \alpha$ **then**

$$s'_{i_s,j} \leftarrow \begin{cases} s'_{i_s,j} + 1 & \text{if } P_{i_s,j}^{\max} > \alpha \quad \forall\, i_s \in \mathbf{I}_s \\ s'_{i_s,j} & \text{if } P_{i_s,j}^{\max} \leq \alpha \quad \forall\, i_s \in \mathbf{I}_s \end{cases}$$

Update cost of $\mathbf{s}'_j$: $c'_{s,j} \leftarrow c'_j + ue_j$
else
if $c'_{s,j} < c_s^*$ **then**
Better feasible solution found: store $s'_{i_s,j}$ and update $c_s^* \leftarrow c'_{s,j}$

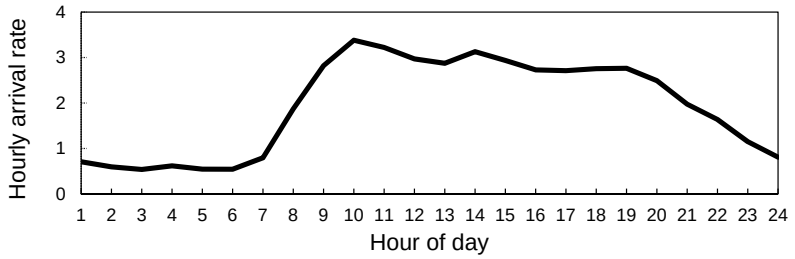
old τ is set to 10 minutes. Per iteration of the algorithm, 2500 replications are performed.

The small-scale system is assumed to operate continuously (we added a warm-up period). The large-scale system is modeled as a terminating system without warm-up (analogous to Feldman et al. [78]). As such, the capacity is decreased to 0 once the time-horizon has elapsed (note that this affects $\Pr(W_t > \tau)$ from time $T - \tau$ onwards).

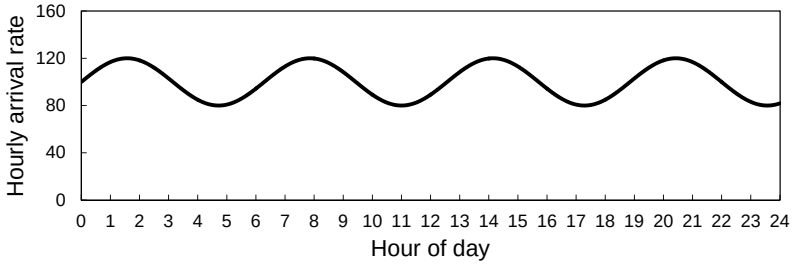
We assume an exhaustive service policy such that servers finish the customer in service before they go off-duty (i.e., they work overtime). We assume a policy with minimum overtime, in which the overtime servers are those that process customers with the lowest remaining service time.

The algorithm's performance for the two examples is first evaluated in an $M_t/M/s_t + M$ context in Section 4.3.1. However, the algorithm re-

mains applicable in an $M_t/G/s_t + G$ context, as the experiments in Section 4.3.2 show. In Section 4.3.3, we compare the proposed ISA(τ) method with the lagged SIPP and MOL heuristics available in the literature. Section 4.3.4 evaluates the algorithm for varying values of τ and for different service policies (exhaustive and preemptive).



(a) Small-scale system: Belgian hospital



(b) Large-scale system: example based on Feldman et al. [78]

Figure 4.3: Arrival rates computational experiment

4.3.1 Exponential service and abandonment times

A comparison of the results for both small- and large-scale settings in an $M_t/M/s_t + M$ setting (see Table 4.3[4]), leads to the conclusion that our algorithm results in staffing levels that indeed meet the desired performance targets in relatively few iterations. The number of iterations needed increases

[4]The CPU times are serve as a indication of the computation time (for general parameter settings); further fine-tuning of the parameters in the simulation model may notably shorten the CPU time.

	Small-scale system Belgian hospital	**Large-scale system Example based on Feldman et al. [78]**
Service rate μ (customers/hour)	2	1
Abandonment rate θ (customers/hour)	0.25	1
Time horizon T	24 hours	
Performance interval Δ_p	1 minute	
Staffing interval Δ_s	15 minutes	
Maximum acceptable wait τ	10 minutes	
Target α	0.1	
Performance constraint	$\Pr(W_t > \tau) \leq \alpha \quad \forall\, t \in \mathbf{t}_p$	
Number of replications per iteration R	2500	

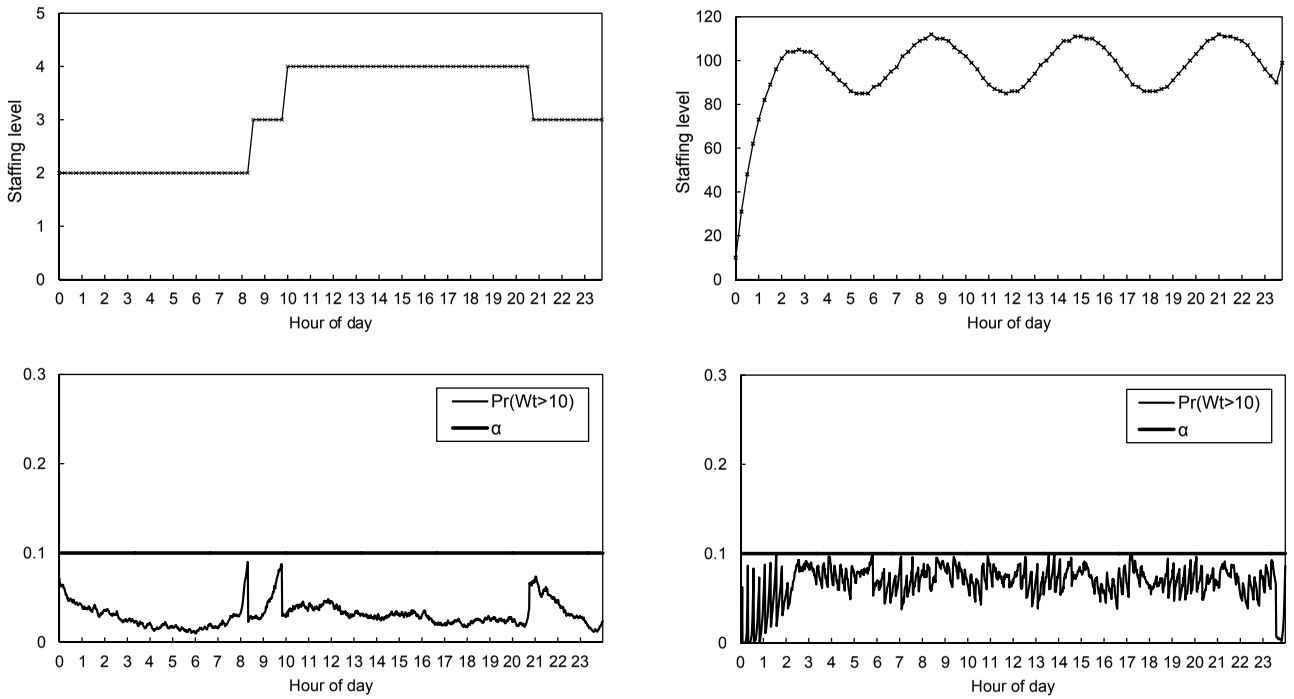
Table 4.2: System parameters

with system size: small-scale systems require less iterations than large-scale systems. Moreover, as the capacity changes are more frequent and larger in size in the large-scale system (see Figure 4.4), so are the performance shocks.

	Small-scale system	Large-scale system
Number iterations PHASE I	6 (0 feasible)	13 (0 feasible)
Number iterations PHASE II	2 (1 feasible)	5 (1 feasible)
$\max_{t \in \mathbf{t}_p}\{P_t\}$	0.0900	0.0992
Staffing cost c_s^*	74.25	2296.00
CPU time (in min)	0.30	1.66

Table 4.3: Results ISA(τ): exponential service and abandonment times

As mentioned in Section 4.2.3, the convergence in the exploration phase can be influenced by changing the scaling factor k in Expression 4.6. Appendix C compares the results for both systems, using alternative scaling factors. As evident from these results, the scaling factor k may affect the number of iterations required, especially for the large-scale system. Moreover, the quality of the final solution does not remain unaffected: the staffing cost may deteriorate if k is not selected appropriately.



(a) Small system

(b) Large system

Figure 4.4: Staffing vector and resulting performance (exponential service and abandonment times)

4.3.2 Lognormal service and abandonment times

One of the major advantages of a simulation-based method is its general applicability, therefore, experiments were repeated assuming an $M_t/G/s_t+G$ setting. Within the set of general distributions, we opt for the Lognormal distribution (as in [34, 38]): service and abandonment times are lognormally distributed with squared coefficient of variation (C^2) equal to 0.5 and 2. As the results in Table 4.4 indicate, the algorithm's effectiveness remains similar. Also, the number of iterations needed by the algorithm does not change substantially.

	Small-scale system		Large-scale system	
	$C^2 = 0.5$	$C^2 = 2$	$C^2 = 0.5$	$C^2 = 2$
Number iterations PHASE I	8	6	24	13
Number iterations PHASE II	2	2	17	7
$\max_{t \in \mathbf{t}_p}\{P_t\}$	0.0964	0.0836	0.0996	0.0992
Staffing cost c_s^*	71.75	74.25	2492.25	2319
CPU time (in min)	0.382	0.302	5.231	2.196

Table 4.4: Results ISA(τ): Lognormal service and abandonment times

4.3.3 Comparison to other staffing heuristics

In this section, the staffing vector obtained by ISA(τ) is compared to some readily implementable staffing heuristics based on the stationary approximations as discussed in Section 4.1.1. The key features of the selected heuristics are summarized in Table 4.5.

We make a distinction between lagged SIPP heuristics (LAGSIPP_CF, LAGSIPP_SRS and LAGSIPP_SIM) and MOL heuristics (MOL_CF, MOL_SRS and MOL_SIM) based on the arrival rate that is inserted into the stationary model. At each time t, lagged SIPP heuristics use the original arrival rate, shifted by the mean service time, $\lambda_{(t-1/\mu)}$. The MOL heuristics derive the arrival rate from the infinite server offered load, m_t^∞. As shown in the table, the stationary models are calculated based on the maximum of the arrival rate over the staffing interval for both approaches (so the resulting staffing levels are rather conservative).

The LAGSIPP_CF and MOL_CF heuristics are based on the available closed form results for stationary $M/M/s$ models (i.e., *without* abandonments). In each interval, the closed form formula for the excess wait probability in a stationary $M/M/s$ model (cf. Gross et al. [102] pp. 66-72) is used to assess the performance corresponding to a staffing level. The smallest staffing level that meets the constraint is selected.

The LAGSIPP_SRS and MOL_SRS heuristics are based on the SRS rule. The value for β is determined using the Garnett delay function [88], as such, these heuristics approximate the $M_t/G/s_t + G$ system by a series of $M/M/s + M$ models.

LAGSIPP_SIM and MOL_SIM heuristics apply the lag SIPP and MOL approximations with general service and abandonment times, by means of a $M/G/s + G$ simulation model. Similar to LAGSIPP_CF and MOL_CF, the probability of excessive waiting is evaluated for various capacity levels, each time selecting the smallest staffing level that meets the constraint.

To allow for a fair comparison with ISA(τ), the value of α needs to be rescaled for the LAGSIPP_SRS and MOL_SRS heuristics (a detailed discussion is given in Appendix D).

Abbreviation	Arrival rate in interval i_s, starting at time $t \in \mathbf{t}_s$	Applied approximation	Staffing by means of
MOL_CF	$\max\{m_j^\infty \mu : j \in [t, t+\Delta_s]\}$	$M/M/s$	Closed form results $M/M/s$
MOL_SRS	$\max\{m_j^\infty \mu : j \in [t, t+\Delta_s]\}$	$M/M/s+M$	Square root staffing rule (using Garnett delay function [88])
MOL_SIM	$\max\{m_j^\infty \mu : j \in [t, t+\Delta_s]\}$	$M/G/s+G$	Simulation of $M/G/s+G$ queue
LAGSIPP_CF	$\max\{\lambda_{(j-1/\mu)} : j \in [t, t+\Delta_s]\}$	$M/M/s$	Closed form results $M/M/s$
LAGSIPP_SRS	$\max\{\lambda_{(j-1/\mu)} : j \in [t, t+\Delta_s]\}$	$M/M/s+M$	Square root staffing rule (using Garnett delay function [88])
LAGSIPP_SIM	$\max\{\lambda_{(j-1/\mu)} : j \in [t, t+\Delta_s]\}$	$M/G/s+G$	Simulation of $M/G/s+G$ queue

Table 4.5: Heuristics available in the literature

Each heuristic is applied to both the small- and large-scale system (cf. Table 4.2) and is compared with the ISA(τ) solution. Figure 4.5 plots the staffing vectors for both systems, assuming exponential distributions in the service and abandonment process. Figures 4.6 and 4.7 show the corresponding probability of excessive waiting for each heuristic.

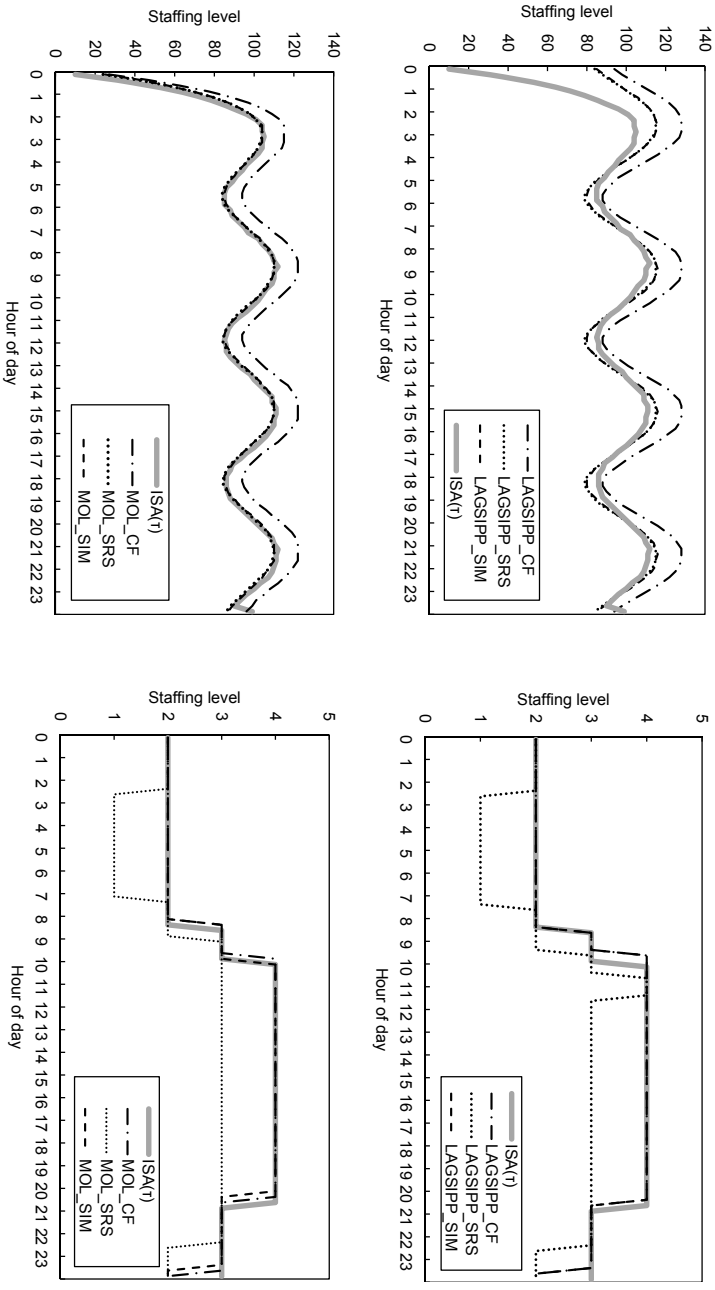
The MOL approximations clearly capture the system dynamics more accurately than the lagged SIPP heuristics. MOL_SRS and MOL_SIM are the most promising: Figure 4.5(a) shows that their staffing levels are similar to ISA(τ), yet, the performance constraint is still violated occasionally (see Figure 4.6(b)). It comes as no surprise that the MOL_SRS and MOL_SIM staffing levels are similar for exponentially distributed service and abandonment processes: in this setting, the offered load used in the SRS_MOL heuristic is exact[5] and in addition the conditions for the Garnett delay function are met (i.e., exponential service and abandonment times and a sufficiently large number of servers). The excess wait probability occasionally exceeds the target though, which might be explained in part by the presence of staffing intervals, and by the fact that the SRS rule is a rule of thumb and therefore provides no guarantee for the performance constraint being met (for MOL_SRS).

The LAGSIPP_CF and MOL_CF heuristics tend to overstaff because abandonments are disregarded in the closed form $M/M/s$ formula (the performance constraint is usually met, due to this overstaffing).

The results for the small-scale setting (given in Figures 4.5(b) and 4.7) show that none of the SRS-based heuristics result in adequate staffing. This might be addressed to the SRS rule performing best for moderate to large offered loads [88], whereas we applied it to a very small-scale system. The closed-form formula results in slightly better staffing vectors, although again, it has the tendency to overstaff as the presence of abandonments is ignored (for this example, the overstaffing remains limited due to the low abandonment rate).

ISA(τ) is the only method that consistently meets the performance tar-

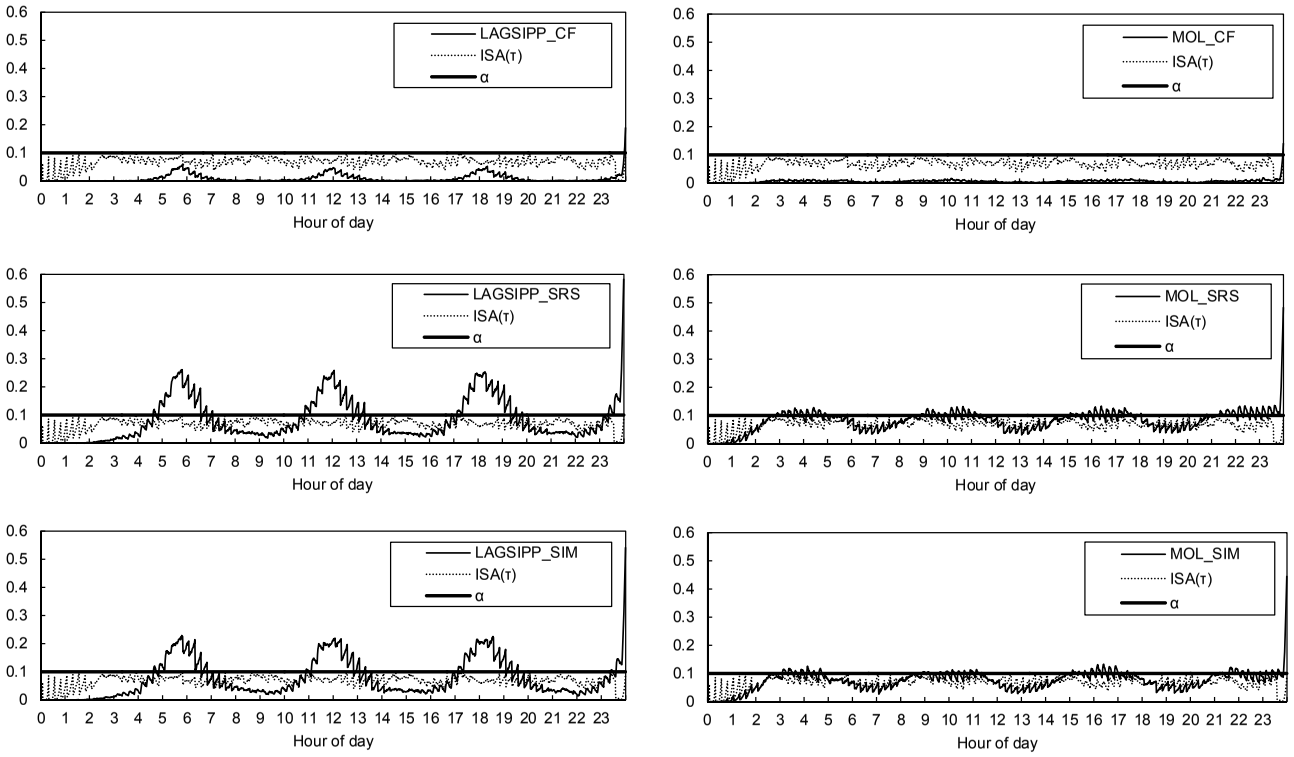
[5]The distribution of number in system in an $M_t/M/s_t + M$ system is identical to that of the infinite server model, if the specific condition holds that the abandonment rate is equal to the service rate [253], as is the case in this setting.



(a) Large-scale system

(b) Small-scale system

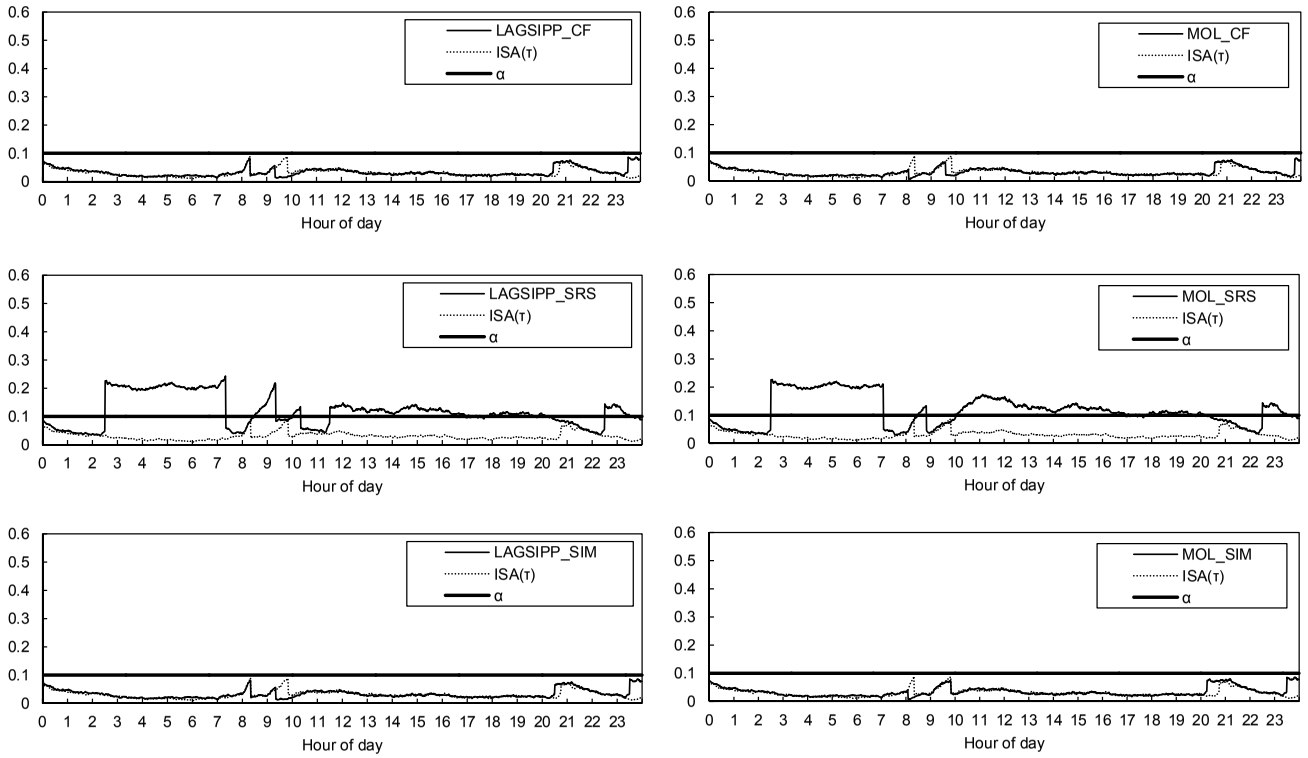
Figure 4.5: Comparison ISA(τ) staffing vs. other heuristics (exponential service and abandonment times)



(a) Comparison ISA(τ) vs. lagged SIPP

(b) Comparison ISA(τ) vs. MOL

Figure 4.6: Probability of excessive waiting (large-scale system, exponential service and abandonment times)



(a) Comparison ISA(τ) vs. lagged SIPP

(b) Comparison ISA(τ) vs. MOL

Figure 4.7: Probability of excessive waiting (small-scale system, exponential service and abandonment times)

	Small-scale system			Large-scale system		
	$C^2 = 0.5$	$C^2 = 1$	$C^2 = 2$	$C^2 = 0.5$	$C^2 = 1$	$C^2 = 2$
Staffing cost c_s						
LAGSIPP_CF	74.00	74.00	74.00	2651.50	2651.50	2651.50
LAGSIPP_SRS	57.00	57.00	57.00	2378.75	2378.75	2378.75
LAGSIPP_SIM	74.00	74.00	74.00	2601.75	2389.75	2479.00
MOL_CF	74.25	74.25	74.25*	2549.50	2549.50	2549.50
MOL_SRS	56.75	56.75	56.75	2288.75	2297.74	2288.75
MOL_SIM	74.00	73.50*	75.00	2529.00	2299.50	2335.00
ISA(τ)	71.75*	74.25	74.25*	2492.25*	2296.00*	2319.00*
Performance $\max_{t \in \mathbf{t}_p}\{P_t\}$						
LAGSIPP_CF	0.0860	0.0860	0.1008	0.1108	0.0588	0.1084
LAGSIPP_SRS	0.2424	0.2428	0.2356	0.6864	0.2608	0.4184
LAGSIPP_SIM	0.0860	0.0860	0.1008	0.1704	0.2288	0.2808
MOL_CF	0.0752	0.0820	0.0864*	0.1520	0.0180	0.0308
MOL_SRS	0.2292	0.2272	0.2356	0.6916	0.1352	0.2240
MOL_SIM	0.0780	0.0860*	0.0704	0.0920	0.1328	0.1260
ISA(τ)	0.0964*	0.0900	0.0836*	0.0996*	0.0992*	0.0992*

Table 4.6: Comparison: ISA(τ) vs. lagged SIPP and MOL (solutions that are feasible w.r.t. the performance constraint are underlined; the cheapest feasible solution is indicated by *)

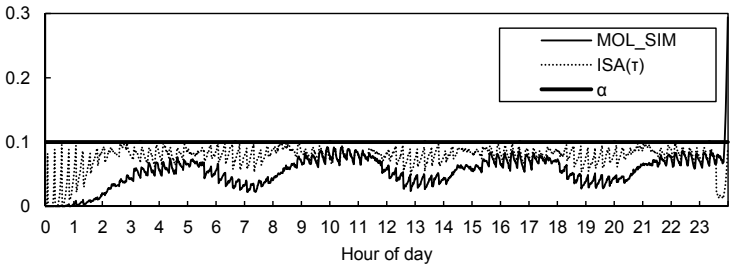
get for the exponential and lognormal settings. Table 4.6 presents for each heuristic the cost of the staffing solution and the maximum probability of excessive waiting encountered over the time horizon[6]. Among the methods that meet the performance constraint, ISA(τ) consistently yields the cheapest solution in the large-scale system. In the small-scale system, MOL_SIM generated a cheaper feasible solution if $C^2 = 1$ (though the cost difference is relatively small). Apart from that, the table reveals that LAGSIPP_CF and MOL_CF frequently result in feasible staffing vectors for different values of C^2, despite the assumptions of exponential service times and no impatience (however, these heuristics may overstaff severely if the abandonment rate is high). The large maximum excess wait probabilities for the lognormal settings of LAGSIPP_SRS and MOL_SRS, on the other hand, suggest that the Garnett delay function should be used with caution if the service and

[6]For the large-scale systems, we disregarded the last τ minutes, because the worse performance is clearly the result of modeling it as a terminating system (with zero capacity, once the time horizon has elapsed).

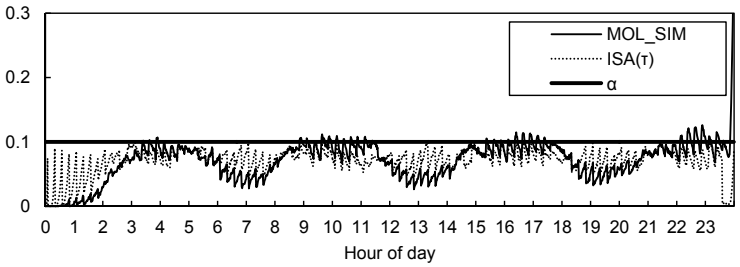
abandonment distributions do not follow an exponential distribution.

The MOL_SIM staffing vector often lies surprisingly close to that of ISA(τ), regardless of the value for C^2. Figures 4.8(a) and 4.8(b) reveal that the corresponding probability of excessive waiting generally is close to the target. This provides support for the MOL approximation in $M_t/G/s_t + G$ systems. However, even these small differences in staffing may cause violations of the performance target, especially for the large-scale system.

Consequently, we may conclude that ISA(τ) is the only heuristic that yields consistent and satisfactory performance, both for small- and large-scale systems, and in particular in settings where the exponential assumptions are not valid.



(a) Large system, $C^2 = 0.5$



(b) Large system, $C^2 = 2$

Figure 4.8: Comparison ISA(τ) staffing vs. MOL_SIM: Probability of excessive waiting (large-scale system, lognormal service and abandonment times)

4.3.4 Impact of the service policy

We discuss the insights revealed by examining the ISA(τ) solutions when using various values for τ and other service policies. Recall that the computational results in the previous sections assumed an exhaustive service policy where overtime servers were selected as the ones with the shortest remaining process time (if any). We found that this service policy plays an important role in staffing problems and that an alternative objective function (that includes the *overtime* cost) may be more appropriate for some service policies.

We define the following service policies, that specify which servers leave the system if capacity decreases:

- PRE: Preemptive service policy. Idle servers (if any) are assumed to leave first, if capacity decreases. Additionally, customers in service may be sent back the queue. These are randomly selected from the set of busy servers.
- EXH(1^{st}): Exhaustive service policy, first-completing-leaves-first. Idle servers (if any) are assumed to leave first, if capacity decreases. Then, the servers with the shortest remaining process time switch to overtime (if needed). This is the service policy applied in Sections 4.3.1 to 4.3.3.
- EXH(rand): Exhaustive service policy, random. The departing servers are selected randomly from the set of idle servers and busy servers (only the selection of busy servers initiates overtime).

The exhaustive service policy (and thus the use of overtime at end-of-shift epochs) is widespread in practice [69], yet, the preemptive policy is most common in the academic literature on nonstationary arrivals (it is the "natural" approach from a computational point of view; [123]). We found only one article that provides a mathematical definition on how to implement an exhaustive policy in an analytical (or simulation) model ([123]; it corresponds to EXH(rand)).

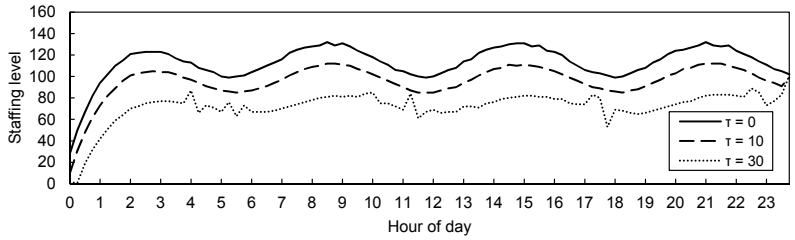
Allowing for overtime has a beneficial effect on the probability of excessive waiting: Ingolfsson [123] and Ingolfsson et al. [121] remark that customers being serviced by an overtime server can be considered as "ejected from the system", because they no longer influence the waiting times of cus-

tomers arriving after the overtime period was initiated. The policies listed above are ordered based on the (potential) number of overtime servers: In the PRE policy, overtime does not occur, EXH(1^{st}) yields a minimum number of overtime servers (with minimum amount of overtime), and EXH(rand) will likely result in a larger number of overtime servers.

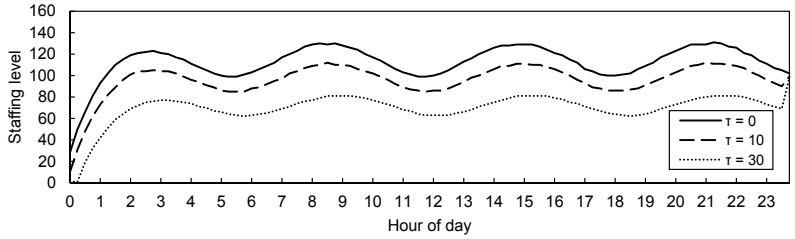
These service policies become particularly relevant when solving staffing problems with Δ_s small compared to τ: Figures 4.9(a)-4.9(c) present the staffing levels generated by ISA(τ)[7] and for varying maximum allowed waiting time ($\tau = 0, 10$, and 30 minutes), for the service policies described above. The performance constraint was met for all vectors in Figure 4.9, as they were obtained by ISA(τ).

For EXH(rand), the exhaustive service policy that results in the highest number of overtime servers (see Figure 4.9(c)), ISA(τ) generates staffing vectors that fluctuate heavily between consecutive staffing intervals. This is because introducing heavy fluctuations in the staffing level may be beneficial in an exhaustive service policy. An example is shown in Figure 4.10: at time t_1, the capacity increases drastically and many (or all) customers in the queue enter service. The capacity drops to almost 0 at t_2 such that a large number of busy servers switch to overtime (and, many customers are "ejected" from the system). At time t_3, the capacity returns to its initial level and due to this increase, customers arriving during the low-capacity period may still enter service without exceeding the waiting time threshold τ (provided that τ is not too small, compared to Δ_s). The key insight is that a sudden capacity increase, if followed by a capacity decrease shortly afterwards, will cause the queue to decrease (or be emptied) without necessarily affecting the staffing cost. This effect is most prominent for service policies that generate large numbers of overtime servers and if τ is large compared to Δ_s (note that this implies higher flexibility in the choice of the staffing levels, as the waiting time may be then affected by the capacity in several staffing intervals). In such settings, care should be taken when solving staffing problems, and the cost of overtime servers should be included

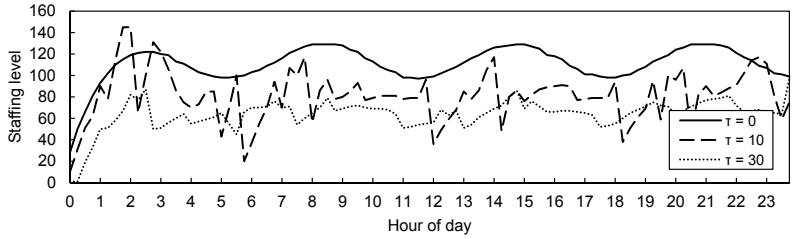
[7]We only consider the large-scale system with exponential service and abandonment times (similar observations hold for the small-scale system, though less clearly observable because the capacity changes are small in size).



(a) Service policy: PRE



(b) Service policy: EXH(1^{st})



(c) Service policy: EXH(rand)

Figure 4.9: ISA(τ) staffing, for varying service policies and τ ($\alpha = 0.1$)

in the objective (especially if the average service time is long).

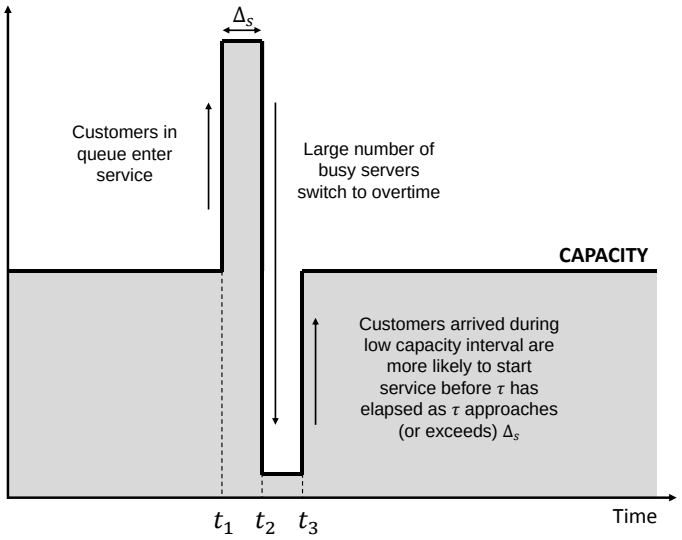


Figure 4.10: Illustration non-smooth staffing for exhaustive service policies

Moreover, we found that if $\tau = 30$, ISA(τ) staffing may fluctuate heavily from one interval to the next, regardless of the service policy. This is likely caused by the fact that the waiting time threshold then exceeds the staffing interval length ($\Delta_s = 15$ min, in our experiments). In theory, the staffing level in any given interval can then be lowered to 0, provided that the capacity in the next interval(s) is sufficiently high (so that customers can enter service within τ minutes after their arrival). Indeed, the performance graphs in Figure 4.11 show that the ISA(τ) solution remains feasible for all service policies, even if $\tau = 30$. This issue can be avoided by including shift constraints in the analysis, as these establish links between the capacity in consecutive periods.

4.4 Conclusions and future research

This chapter suggests an extension of the simulation-based Iterative Staffing Algorithm (ISA) proposed by Feldman et al. [78] as a method to set staffing levels in service systems with nonstationary demand. Our extension —

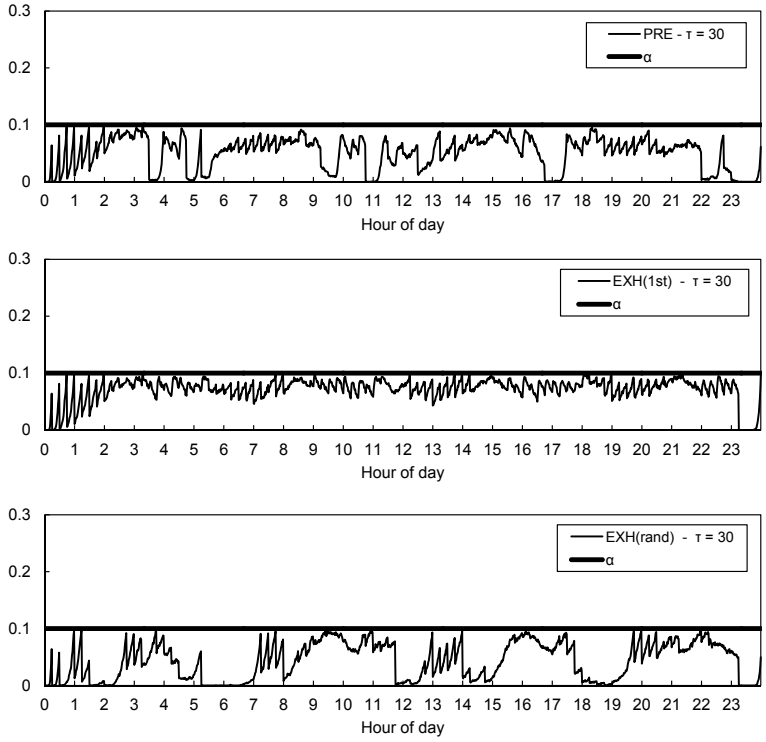


Figure 4.11: ISA(τ) probability of excessive waiting for $\tau = 30$ minutes, for different service policies.

called ISA(τ)— enables to measure performance based on the probability of excessive waiting, instead of the common focus on delay probability as a performance metric. Moreover, it takes into account the sensitivity of small scale systems to changes in the staffing levels, and the presence of staffing intervals. Meanwhile, the advantages of the traditional ISA (namely general applicability, automatic validation) remain valid.

Experiments illustrate that ISA(τ) is both effective and efficient in determining staffing requirements for small-scale and large-scale systems. It consistently outperforms heuristics based on stationary approximations, in particular for settings in which the service and abandonment processes are not exponentially distributed. In general, the efficiency of the algorithm tends to depend on its parameters (both the amplification factor in Expres-

sion 4.6 and the stop criterion can be tuned), and the size of the system (larger systems require more computation time).

Given that ISA(τ) is both effective and efficient in detecting required capacities, and requires no specific tools other than simulation, we are confident that the method offers opportunities to support decisions in practice. The methodology is applicable to any setting in which demand is nonstationary, and in which the decision maker relies primarily on capacity to ensure adequate customer service (such as emergency departments, retail stores, or small- to medium-scale call centers).

Our results in Section 4.3.4, however, indicate that ISA(τ) provides non-smooth (but feasible) staffing levels if the maximum acceptable wait is large compared to the staffing interval length. In that case, introducing shift constraints may provide a means to facilitate the optimization process, because it links the staffing vectors of consecutive intervals. Moreover, the analysis revealed that the service policy impacts the customer waiting times in such a way that it can strongly affect the outcome of staffing and scheduling methods. The exhaustive service policy —which is not common in the academic literature on nonstationary arrivals, but often relevant in practice— may give rise to counter-intuitive staffing solutions. Adding constraints or altering the objective function (e.g., by including overtime cost) should allow to accommodate this issue: this calls for a further exploration of the impact and modeling issues related to the exhaustive service policy and the cost of overtime.

Chapter 5

A branch-and-bound algorithm for shift scheduling with nonstationary demand

5.1 Introduction

Many shift scheduling algorithms presume that the staffing levels, required to ensure a target customer service, are known in advance: the shift scheduling step then boils down to fitting the min cost shift schedule to the requirements. Determining these staffing requirements, however, is nontrivial at best, particularly in systems with nonstationary arrival. Moreover, this "two-step" approach may result in a suboptimal schedule [122].

This chapter presents an *integrated* approach to the shift scheduling problem with a nonstationary (i.e., time-varying stochastic) arrival process: different staffing combinations are explored using implicit enumeration, which allows to efficiently estimate the minimum cost shift schedule subject to a service level constraint (the probability that the customer waiting time violates a critical level should not exceed a user-defined target). The algorithm is flexible in the sense that it does not rely on any specific methodology to evaluate the customer service implied by a given shift schedule. We opted to use simulation in our experiments, because (1) it requires

virtually no restrictions on the assumptions regarding arrival and service process, (2) it allows us to include real-life complexities of which the impact on customer service cannot easily be estimated analytically, such as customer impatience (abandonments) and the exhaustive service policy (which implies that servers work overtime to finish the customer in service at the time their shift ends), and (3) it allows us to tune the accuracy by changing the number of replications in the simulation model.

The algorithm specifically targets service systems with limited opening hours (so-called *terminating* systems, see Law and Kelton [158]), and is especially suited for systems with a limited number of operators (such as banks, retail stores, or small call centers). It contributes to the existing literature by proposing straightforward, easy-to-implement rules to efficiently explore the solution space (as opposed to the more complex and time-consuming approach of Atlason et al. [13, 14]).

Section 5.2 gives a brief discussion of the related literature. Section 5.3 presents the formal problem statement. A detailed description of the algorithm is provided in Section 5.4. Section 5.5 discusses the computational experiment, and concluding remarks are provided in Section 5.6.

5.2 Related literature

Shift scheduling for systems with nonstationary arrivals has received relatively limited attention in the academic literature. The two-step approach, which fits minimum cost shift schedules to predefined staffing requirements, is by far the most common (see Thompson [230, 232], Sinreich and Jabali [218], and Izady and Worthington [125], among others). The main problem, however, is that the staffing levels required to ensure a target customer service level are not straightforward to determine. Figure 5.1 illustrates this two-step approach, for a fictive problem setting with a sinusoidal arrival pattern that displays two peaks per day. The staffing requirements are determined by the ISA(τ) algorithm (see Chapter 4), and are considered as a strict lower bound on the required capacity. In the shift requirements, which are derived by solving Dantzig's set covering problem [60], slack capacity is added to meet the shift constraints.

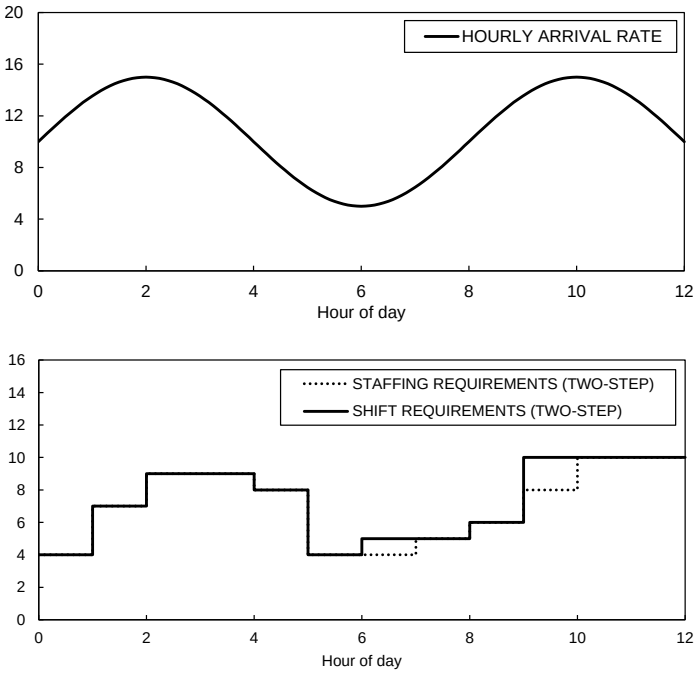


Figure 5.1: Two-step approach: arrival rate, staffing requirements and shift requirements

This two-step approach, however, may result in suboptimal shift schedules [120, 110, 111] because several staffing solutions might exist that lead to shift schedules with substantially varying costs, and because the staffing problem is usually solved heuristically. Figure 5.2 illustrates this suboptimality: it shows the shift requirements (as derived by the two-step approach, and identical to Figure 5.1), and the estimated optimal shift requirements (generated by the method presented in this chapter). As can be seen in the figure, the optimum shift schedule requires notably less capacity. Alternatively, shift scheduling can be done directly based on the time-varying arrival rates [120, 155, 46, 109]. These approaches avoid the suboptimality that arises by decomposing the problem into two steps. Yet, including quality of service constraints in the shift optimization is not straightforward, hence authors commonly resort to simplifying assumptions (e.g., exponential service and abandonment times).

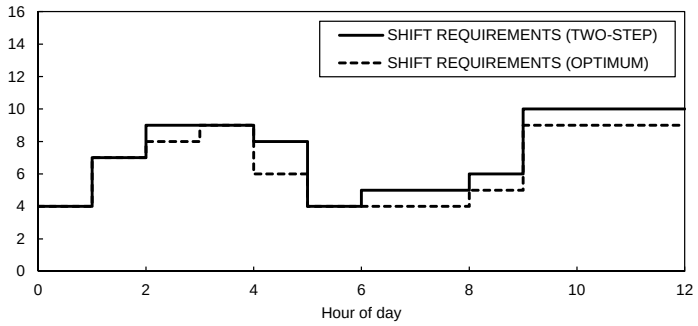


Figure 5.2: Suboptimality of the two-step approach: illustration

Our research is closely related to the work of Ingolfsson et al. [120, 122] and Atlason et al. [13, 14]. These articles suggest algorithms to determine low-cost shift schedules with a service level constraint on customer waiting time. Ingolfsson et al. [120] evaluate schedule performance by numerical integration of the forward differential equations for $M_t/M/s_t$ queues and apply a genetic algorithm to search for good schedules. Ingolfsson et al. [122] apply a heuristic cutting-plane algorithm and use the randomization method for evaluating schedule performance [90, 123, 121], which is computationally less expensive but yields similar accuracy [121]. Atlason et al. [13, 14] suggest a cutting plane method that uses simulation to evaluate customer service, and add cuts based on the estimated (pseudo)gradients of the service level function. This requires substantial computational effort. Atlason et al. [14] show that their algorithm converges towards an optimal solution as the number of replications grows large; in contrast, both Ingolfsson et al. [120] and Ingolfsson et al. [122] are heuristic approaches, that do not guarantee an optimal solution.

The approach developed in this chapter is easier to implement than the one proposed in Atlason et al. [13, 14]. Though our approach cannot strictly guarantee the optimum in the exhaustive setting, it will converge to the optimal solution in systems with a preemptive service policy (where service can be interrupted and the customer in service rejoins the queue), as the number of replications grows to infinity.

5.3 Problem statement and notations

The notations used throughout this chapter are in line with the previous chapters (for ease of reference, Table 5.1 provides a notation list).

The main objective is to estimate an optimal shift schedule, such that the target customer service is achieved at minimum cost. The cost is measured in man-hours. In line with the related literature [78, 122, 44, 125], customer service is measured by the *virtual waiting time* W_t at given time instants t (we use the same simulation-based method as in Chapter 4).

We then require the following hard constraint to be met:

$$\Pr(W_t > \tau) \leq \alpha \quad \text{for all } t \in \mathbf{t}_p\,, \tag{5.1}$$

with τ the maximum allowed waiting time, and α the target probability of excessive waiting. The validity of this constraint is checked by simulation. Note that for $\tau = 0$, Expression (5.1) corresponds to the delay probability.

Capacity changes can only take place at specific points in time, i.e., at the start of a *staffing interval*. Staffing intervals have length Δ_s. The set of staffing interval indices is $\mathbf{I}_s = \{1, \ldots, I_s\}$ with $I_s \equiv T/\Delta_s$. $\mathbf{t}_s = \{0, \Delta_s, 2\Delta_s, \ldots, T - \Delta_s\}$ contains the staffing interval start times, for all $i_s \in \mathbf{I}_s$ (with $\mathbf{t}_s \subseteq \mathbf{t}_p$). Let vector $\mathbf{s} = \{s_1, ..., s_{I_s}\}$ represent the staffing vector, containing the number of operators in each staffing interval.

Assume that W different pre-defined shift types exist, that differ in terms of shift duration (e.g., 4-hour or 8-hour shifts), start time (e.g., morning shift, afternoon shift or night shift), and the timing of breaks (e.g., shifts with standard breaks versus split shifts that use 2- to 4-hour breaks [18]). For any staffing vector $\mathbf{s}$, the min-cost shift solution can be determined by solving the following basic set covering problem (as introduced in Dantzig [60]):

$$\min \qquad \sum_{j=1}^{W} c_j w_j \tag{5.2}$$

$$\text{s.t.} \quad \sum_{j=1}^{W} a_{j,i_s} w_j \geq s_{i_s} \quad \forall i_s \in \mathbf{I}_s \tag{5.3}$$

$$w_j \geq 0 \text{ and integer } \forall j = 1, \ldots, W \tag{5.4}$$

Notation		
$\mathbf{I}_p$	:	Set of performance intervals
i_p	:	Performance interval index, $i_p \in \mathbf{I}_p$
$\mathbf{I}_s$	:	Set of staffing intervals
i_s	:	Staffing interval index, $i_s \in \mathbf{I}_s$
t	:	Time index, with $t \in [0, T]$
$\mathbf{t}_s$	:	Set of staffing interval start times
$\mathbf{t}_p$	:	Set of performance interval start times
W_t	:	Waiting time of a virtual customer arriving at $t \in \mathbf{t}_p$
τ	:	Maximum acceptable waiting time
$\Pr(W_t > \tau)$	:	Probability of experiencing an excessive wait, upon arrival at time t
α	:	Target w.r.t. $\Pr(W_t > \tau)$
λ_t	:	Arrival rate, as a function of t
μ	:	Service rate
θ	:	Abandonment rate
$\mathbf{A}$	:	Shift specification matrix, with $a(j, i_s) = 1$ if a server is active during interval i_s in shift type j; $a(j, i_s) = 0$ otherwise
$\mathbf{s}$	:	Staffing vector, $\mathbf{s} = \{s_1, ..., s_{I_s}\}$
$\mathbf{w}$	:	Shift vector; number of servers assigned to each shift ($\mathbf{w} = \{w(1), ..., w(W)\}$)
$\mathbf{s}_w$	:	Shift vector; number of servers available during each staffing interval ($\mathbf{s}_w = \{s_{w,1}, ..., s_{w,I_s}\}$)
c_s	:	Cost of $\mathbf{s}$, staffing cost
c_w	:	Cost of $\mathbf{w}$, shift cost
c_{tot}	:	Simulated total cost of $\mathbf{w}$, including overtime cost
$\mathbf{s}^{init}$	:	Initial feasible staffing vector
$\mathbf{w}^{init}$	:	Initial feasible shift vector
$\mathbf{s}_w^{init}$	:	Initial feasible staffing vector
c_w^{init}	:	Cost of $\mathbf{w}^{init}$
$\mathbf{w}^*$	:	Best feasible shift vector found so far
$\mathbf{s}_w^*$	:	Best feasible shift vector found so far
c_w^*	:	Cost of $\mathbf{w}^*$
c_{tot}^*	:	Total cost of $\mathbf{w}^*$
$\mathbf{s}^{\mathsf{LB}}$	:	Lower bound on staffing requirements
$\mathbf{s}^{\mathsf{UB}}$	:	Upper bound on staffing requirements
$\phi(\mathbf{s}_w)$	:	Equal to 1 if the simulated shift vector is feasible, 0 otherwise.
t^e	:	Time at which first excessive wait occurs, $t^e \in \mathbf{t}_p$
i_s^e	:	Latest staffing interval whose capacity can be modified to achieve acceptable performance at time t^e

Table 5.1: Chapter 5: notations

The objective function denotes the total shift cost, with c_j the cost of shift j (expressed in man-hours). In constraint (5.3), the indicator a_{j,i_s} equals 1 if interval i_s is an active period in shift j and equals 0 otherwise. Constraint (5.4) imposes non-negativity on the shift solution vector $\mathbf{w} = \{w_1, ..., w_W\}$, that defines how many workers are assigned to each shift type. The actual number of operators implied by a given shift vector $\mathbf{w}$ is expressed as $\mathbf{s}_w = \{s_{w,1}, \ldots, s_{w,I_s}\}$. Note that different $\mathbf{w}$ may give rise to the same $\mathbf{s}_w$, and that $\mathbf{s}_w$ will tend to differ from $\mathbf{s}$, as the shift schedule often introduces slack on the first constraint in Problem (5.2-5.4).

The overall objective is to minimize the shift cost c_w, while ensuring that the related shift vector w satisfies the performance constraint in Expression (5.1). The abandonment cost is not included in the objective, but instead is influenced implicitly through the performance constraint: as abandonment behavior will increase as the waiting times grow, τ should be small compared to $1/\theta$ if abandonments are to be avoided.

The exhaustive service policy implies that servers will work overtime at the time their shift ends, to finish the ongoing service instance (if any). As such, customers cannot be transferred between servers. Note that this does not completely match the exhaustive service policy applied in Atlason et al. [14], which only allows for overtime when the overall scheduled capacity decreases (i.e., when the servers that go off duty are not replaced by new servers).

5.4 Branch-and-bound algorithm

In this section, we develop a branch-and-bound algorithm for shift scheduling with nonstationary arrivals. Section 5.4.1 discusses how the search tree is constructed. Section 5.4.2 describes in detail how this tree is explored, and which rules are applied to guide the search procedure.

5.4.1 Tree structure

The construction of the tree requires the following three staffing vectors as input:

- an initial feasible solution $\mathbf{s}^{\mathsf{init}}$: any staffing vector that satisfies the performance constraint qualifies as initial feasible solution. A tighter initial feasible solution, however, speeds up convergence. The corresponding min-cost shift vector, $\mathbf{w}^{\mathsf{init}}$ (with cost c_w^{init}), is obtained as the integer programming solution to Problem (5.2-5.4).

- a lower bound vector $\mathbf{s}^{\mathsf{LB}}$: this vector contains the lower bound on the staffing requirements for each interval $i_s \in \mathbf{I}_s$. Any staffing vector with capacity smaller than $\mathbf{s}^{\mathsf{LB}}$ in at least 1 interval, can never be feasible. Appendix E details how to obtain $\mathbf{s}^{\mathsf{LB}}$.

- an upper bound vector $\mathbf{s}^{\mathsf{UB}}$: all solutions for which $s_{i_s} > s_{i_s}^{\mathsf{UB}}$ in at least one staffing interval yield a staffing cost that exceeds c_w^{init}, and should not be considered. Appendix E details how to obtain $\mathbf{s}^{\mathsf{UB}}$.

An illustration of the tree structure is presented in Figure 5.3, for $I_s = 3$. Each node in the tree represents a staffing vector $\mathbf{s}$, with corresponding staffing cost c_s. The root node of the tree is initialized to $\mathbf{s}^{\mathsf{LB}}$ (as staffing vectors with capacity smaller than $\mathbf{s}^{\mathsf{LB}}$ in at least 1 interval are infeasible, they need not be considered in the search tree).

Starting from the root node, $\mathbf{s}$ is increased throughout the search tree. Each level in the tree is denoted by its depth $d = 0, \ldots, I_s$ ($d = 0$ represents the depth of the root node). Child nodes are generated from a parent node by adding capacity to a given staffing interval (see Figure 5.4): child nodes at level $d+1$ differ in capacity from the parent node in staffing interval $d+1$, where the staffing level takes values between its lower bound, s_{d+1}^{LB} and its upper bound, s_{d+1}^{UB}. The staffing levels in the other intervals are identical to those of the parent node.

The total number of nodes in the tree (denoted by $\mathcal{S}$) is equal to:

$$\mathcal{S} = \prod_{i_s=1}^{I_s} \left(s_{i_s}^{\mathsf{UB}} - s_{i_s}^{\mathsf{LB}} + 1 \right). \tag{5.5}$$

As explained in Section 5.4.2, defining the search tree in terms of staffing vectors enables an efficient exploration of the solution space. For each staffing vector $\mathbf{s}$, the corresponding min-cost shift solution $\mathbf{w}$ can be

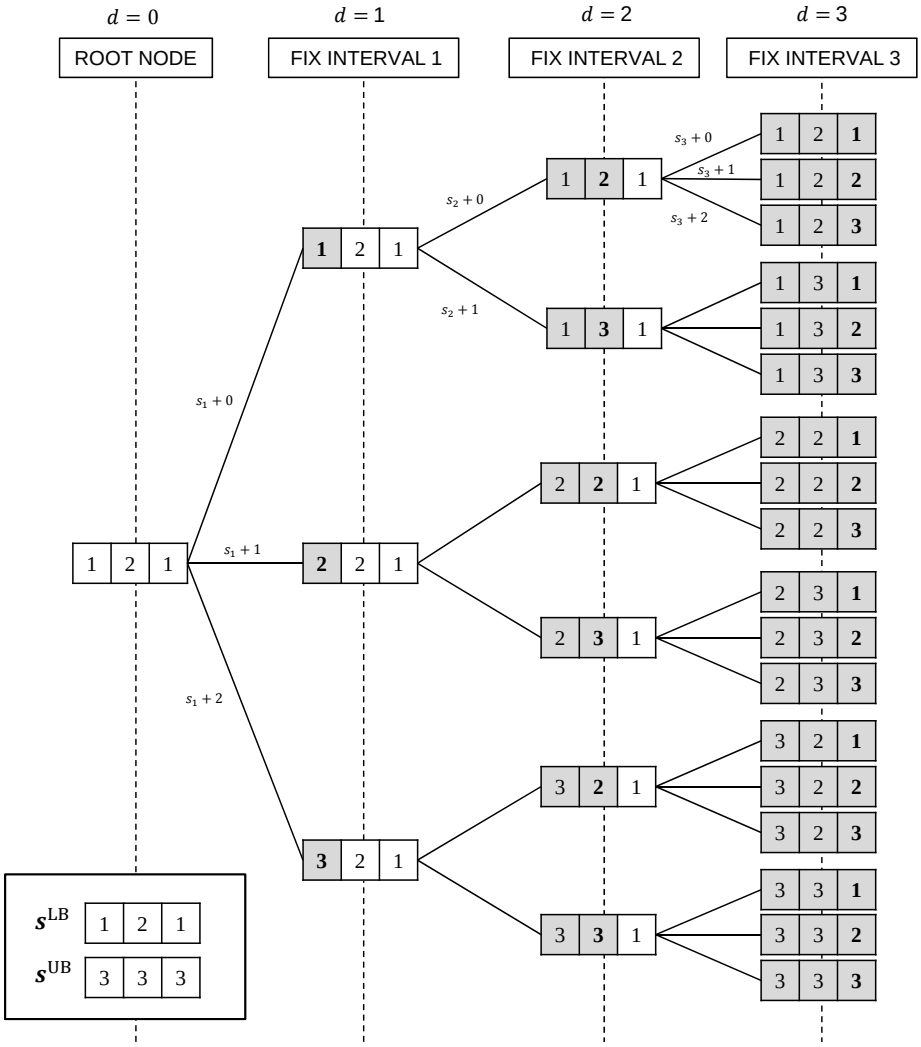


Figure 5.3: Example tree structure ($I_s = 3$)

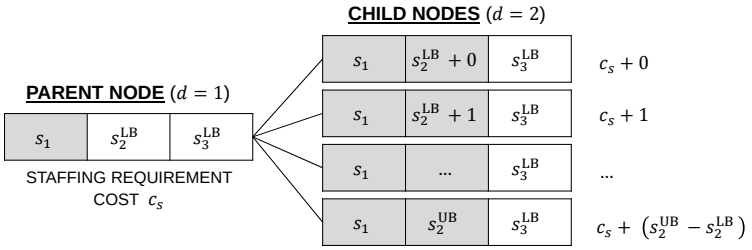


Figure 5.4: Illustration: branching to a lower level ($I_s = 3$)

retrieved by solving Problem (5.2-5.4). Note that the **s** vectors themselves are not checked for feasibility with respect to the performance constraint: only the feasibility of **w** is relevant. By implicitly enumerating all staffing vectors, the algorithm avoids the suboptimality that is inherent in the traditional two-step approach.

5.4.2 Node exploration

For any given parent node, child nodes are considered in increasing order of c_s (i.e., from top to bottom, in Figure 5.3). The tree is explored in a depth-first manner: after checking a node at depth d, the algorithm branches to the lowest cost child node at levels $d+1, d+2, \ldots$ etc. If the lowest level is reached ($d = I_s$) and all child nodes of the current parent node have been explored, we *backtrack*: the algorithm then returns to the previous level and continues with the next unexplored node in the tree. Note that in Figure 5.3, the top child node at level $d+1$ duplicates the parent node at level d; these duplicates are shown for completeness and are not explored.

To limit the number of nodes for which we need to effectively simulate the customer service level, we implement rules to *fathom* nodes. A node is fathomed if it is discarded from the search procedure, along with all its underlying child nodes. Throughout the algorithm, the best (feasible) shift vector found so far is stored ($\mathbf{w}^*$, with shift cost c_w^*). At the start of the algorithm, $\mathbf{w}^*$ is initialized to $\mathbf{w}^{\mathsf{init}}$.

Every node in the tree is evaluated according to the rules summarized in Figure 5.5. Sections 5.4.2.1, 5.4.2.2 and 5.4.2.3 describe the computationally

inexpensive rules (Fathom[c_s], Fathom[c_w], Fathom[$c_w^{\mathcal{R}}$], and Fathom[i_s^e]) used in steps 1-3 to identify nodes that can be fathomed. Only shift vectors that cannot be fathomed in these steps, are simulated in step 4. Based on the simulation outcome, two additional fathoming rules (Fathom[$\mathbf{w}^*$] and Fathom[i_s^e]) are applied to further constrain the solution space.

5.4.2.1 Step 1: Evaluate staffing cost c_s

For any node **s**, we first evaluate its staffing cost c_s: if $c_s \geq c_w^*$ (with c_w^* the best *shift* cost so far), then node **s** can be fathomed along with its underlying nodes and all unexplored child nodes from the same parent node. Indeed, all child nodes of **s** have a staffing cost which is at least as large as c_s (as illustrated in Figure 5.4), so their corresponding shift cost cannot be smaller than c_w^*. As nodes at a given level are explored in increasing order of c_s, the same is valid for the remaining unexplored child nodes with the same parent node as **s**. The algorithm then proceeds to the next unexplored node in the tree: this can be a node at depth $d-1$ along the same branch as the parent node, or a node higher in the tree (if backtracking takes place).

This rule is referred to as Fathom[c_s]. Due to its low computational effort, it is used as a first criterion to eliminate parts of the solution space that cannot contain an optimum.

5.4.2.2 Step 2: Evaluate shift cost c_w

If **s** could not be fathomed in step 1, the minimum shift cost c_w is determined. We first solve the LP relaxation of Problem (5.2-5.4); let's denote its shift cost by $c_w^{\mathcal{R}}$. If $c_w^{\mathcal{R}} \geq c_w^*$, then node **s** is fathomed along with its underlying nodes, and all unexplored child nodes from the same parent node (the argument is analogous to the one presented in step 1). As in step 1, the algorithm proceeds to the next unexplored node in the tree. Only when $c_w^{\mathcal{R}} < c_w^*$, Problem (5.2-5.4) is solved with the integrality constraints included; when $c_w \geq c_w^*$, again node **s** is fathomed along with its underlying nodes, and all unexplored child nodes from the same parent node. These fathoming rules are referred to as Fathom[$c_w^{\mathcal{R}}$] and Fathom[c_w].

A limitation of our model is that it selects only one min-cost shift

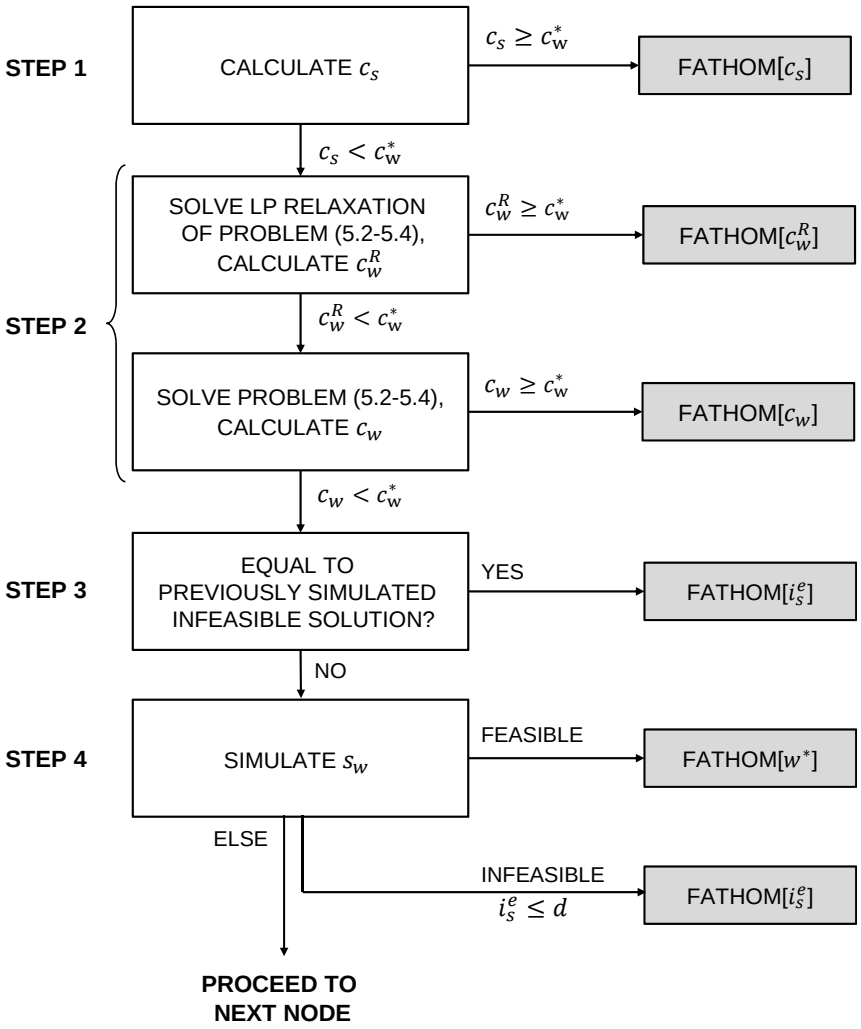


Figure 5.5: Node exploration

vector in each node, as such, possible alternative optima to Problem (5.2-5.4) are not accounted for. In systems with an exhaustive service policy, the start and end times of shifts impact the performance estimates. Alternative shift vectors with identical cost may result in slightly different performance estimates in such a setting (even if the capacity profile $\mathbf{s}_w$ is identical over the day), which could cause the algorithm to miss the optimum. This limitation especially holds for highly utilized systems with long service times, because the exhaustive service policy is most prominent in such settings.

5.4.2.3 Steps 3 and 4: Check if w was simulated before and/or simulate w

Different **s** vectors can result in identical **w** vectors. As such, it is plausible that a given **w** vector with $c_w < c_w^*$ has already been simulated at a previous node. As simulations can be computationally expensive, we store each previously simulated infeasible **w** vector in a set (denoted by **B**), along with information on the first time instant at which the performance constraint was violated:

$$t^e = \min\{t \in \mathbf{t}_p : \Pr(W_t > \tau) > \alpha\}\,. \tag{5.6}$$

If $\mathbf{w} \in \mathbf{B}$ for a given staffing vector **s**, the t^e value allows to detect other infeasible staffing vectors, at least in systems with limited opening hours. Indeed, to attain an acceptable waiting time $W_{t^e} \leq \tau$, a customer arriving at time t^e needs to enter service at $t^e + \tau$ at the latest. If we let i_s^e denote the staffing interval that contains $t^e + \tau$, all $\mathbf{s}'$ for which $s'_{i_s} \leq s_{i_s}$ for all $i_s \in \{1, \ldots, i_s^e\}$ are infeasible as well, irrespective of the capacity in intervals $i_s > i_s^e$ (note that this does not necessarily hold in nonterminating systems). This observation is used to define a last fathoming rule, termed Fathom$[i_s^e]$. Consider an infeasible node at level d, with corresponding i_s^e. Three cases can then be distinguished; these are illustrated in the example provided in Figure 5.6 (for $I_s = 5$):

1. $i_s^e < d$ (see Figure 5.6a). In that case, the parent node at depth i_s^e can be fathomed and the algorithm proceeds with the next unexplored node at level i_s^e.

2. $i_s^e = d$ (see Figure 5.6b). In that case, the node at depth d can be fathomed and the algorithm proceeds with the next unexplored node at level d.
3. $i_s^e > d$ (see Figures 5.6c and 5.6d). In that case, we branch to the next unexplored child node at level i_s^e.

As a result, the algorithm each time augments the capacity in the interval that causes the performance constraint to be violated, i_s^e. Evidently, any violation might also be solved by increasing capacity in prior intervals $i_s < i_s^e$. These solutions will be encountered later in the algorithm, when the algorithm backtracks to levels $d < i_s^e$.

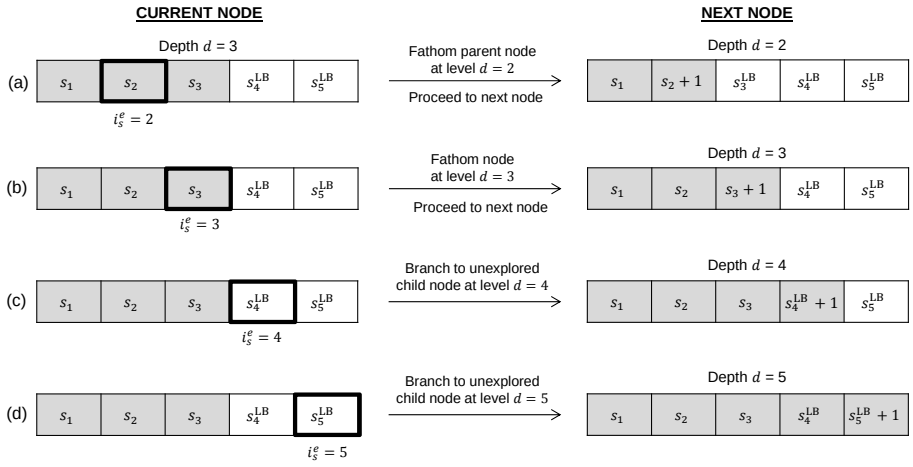


Figure 5.6: Example fathoming and branching based on infeasibility (5 staffing intervals, $d = 3$)

Note that the Fathom$[i_s^e]$ rule is particularly straightforward to apply given that the search tree is defined in terms of **s** vectors and that the rule can be applied to vectors that have been simulated in one of the previous iterations *or* during the current iteration.

5.5 Results

The approach described in Section 5.4 is tested on a set of 972 problem instances. All experiments are performed on an Intel I7 3.40 GHz computer, with 8 GB RAM. The experimental setup is described in Section 5.5.1. Section 5.5.2 discusses the algorithm's computational performance with respect to the number of nodes explored, and the improvement in the shift cost obtained with respect to the initial solution.

5.5.1 Experimental setting

Table 5.2 contains the parameter settings of the test set. We assume that the service system is open 12 hours per day and that the arrival rate follows a sinusoidal pattern with 2 peaks per day, fluctuating around the average rate $\bar{\lambda}$:

$$\lambda_t = \bar{\lambda}\left(1 + RA \sin\left(\frac{2\pi t}{8}\right)\right)$$

where RA denotes the relative amplitude of the arrival rate, and with t expressed in hours. Note that our method does not require the arrivals to follow a sine function; so any other arrival function could be used instead. The service and abandonment distributions are assumed to be of the same type in each of the test instances: either both are exponential ($C^2 = 1$), 2-phase Erlang ($C^2 = 0.5$), or 2-phase Coxian ($C^2 = 2$).

Parameter	**Parameter values**
Service rate μ (customers/hour)	$\{1, 2, 4\}$
Offered load $\bar{\lambda}/\mu$	$\{5, 10, 15\}$
Relative amplitude arrival rate RA	$\{0.5, 1\}$
Abandonment rate θ (customers/hour)	$\{0, \mu\}$
Max wait τ (min)	$\{0, 10, 20\}$
Squared coefficient service and abandonment times (C^2)	$\{0.5, 1, 2\}$
Staffing interval Δ_s (min)	$\{240, 120, 60\}$
Performance interval Δ_p (min)	5
Number replications per simulation R	2500
Target α	0.2

Table 5.2: Experimental setting

The shift sets are provided in Appendix F. Each shift is 4, 6, or 8 hours long and may include a one-hour break. This yields a set of 5 shifts for $\Delta_s = 240$ min, a set of 12 shifts for $\Delta_s = 120$ min, and a set of 45 shifts for $\Delta_s = 60$ min (the latter is identical to the shift set of Ingolfsson et al. [122]). The algorithm is terminated if an estimated optimal solution has not been found after 25,000 nodes have been simulated in the tree exploration phase.

5.5.2 Algorithm performance

In our computational experiments, we select $\mathbf{s}^{\mathsf{init}}$ by means of the ISA(τ) algorithm [65], a staffing heuristic which ensures a fairly tight and feasible staffing solution[1]. Table 5.3 contains statistics on the number of simulation runs needed to find the initial feasible solution and lower bound ("preprocessing" phase). It reveals that the initial feasible solution and the lower bound can be derived with a very small number of simulations.

	Min	Median	Max	Average
Simulations for initial feasible solution	5	16	36	16
Simulations for lower bound	6	18	117	26

Table 5.3: Statistics on the preprocessing phase, over all test instances

Figures 5.7(a) and (b) confirm these findings; they show the number of nodes explored with low computational effort (steps 1 to 3 in Figure 5.5) and high computational effort (step 4 in Figure 5.5), for each problem instance that could be solved to optimality. Figure 5.7(a), that presents the number of nodes explored as a percentage of the total solution space (given by Expression 5.4.1). It shows that the algorithm is efficient: only a minor percentage of the nodes in the solution space are explored during the algorithm. Figure 5.7(b) depicts the absolute numbers, showing more explicitly that the number of nodes requiring simulation is only a fraction of the nodes that are explored with low computational effort (i.e., most observations lie below the diagonal).

[1]The algorithm stops when the solution was already evaluated before (we thus use only one of the stop criteria discussed in Chapter 4)

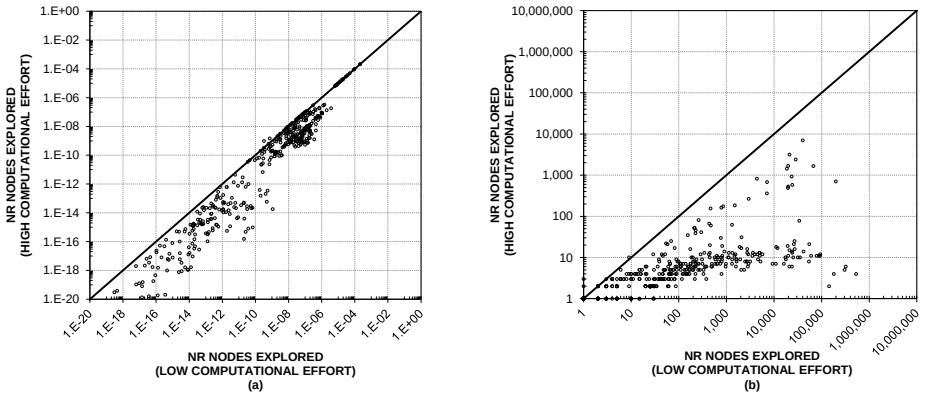


Figure 5.7: Number of explored nodes fathomed by low and high effort fathoming rules, (a) as a percentage of the total solution space, (b) in absolute values.

Figure 5.8 analyzes the computational effort in the tree exploration phase (measured by the number of nodes requiring simulation) versus the improvement in cost obtained with respect to the initial feasible solution. This cost improvement is determined by:

$$\text{Cost improvement} = \frac{(c_w^{\text{init}} - c_w^*)}{c_w^{\text{init}}}. \tag{5.7}$$

The top row of Figure 5.8 shows that in 15.74% of test instances, the algorithm terminates after 25,000 simulation runs (so, without a guarantee that there is no better solution to be found). It appears that the size of the solution space is a decisive factor here. All these instances allowed for 12 staffing intervals ($\Delta_s = 60$). Moreover, as detailed in Table 5.4, the performance decreased as the relative amplitude of the arrival rate and/or the offered load increased (which implies that higher staffing will likely be needed to satisfy the customer service constraint).

The remaining 84.26% of instances were solved to optimality, as summarized in the bottom matrix of Figure 5.8. Table 5.5 gives further details on these instances, analyzing the performance of the algorithm across different parameter settings. We compare only instances that were solved to optimality for each value of a particular parameter (all else equal); the last

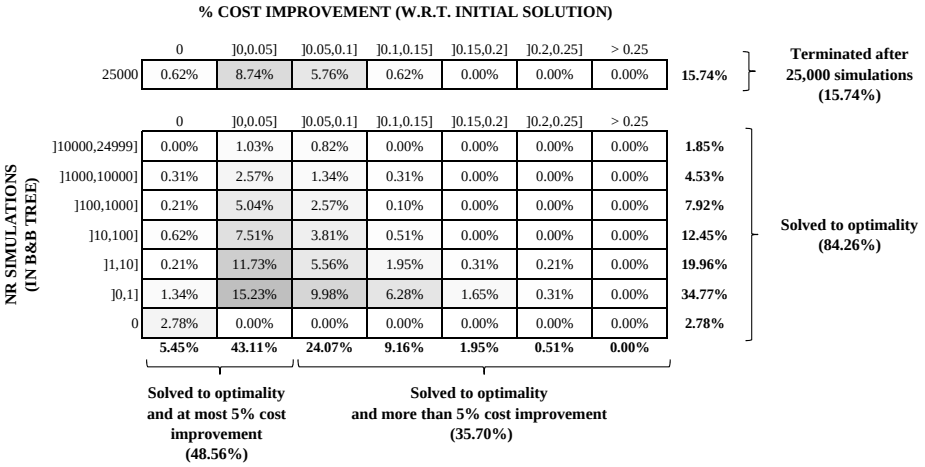


Figure 5.8: Simulation runs performed in branch-and-bound tree vs. percent improvement over the initial solution

	Offered load		
	5	10	15
RA = 0.5	96%	59%	35%
RA = 1	78%	30%	19%

Table 5.4: Percentage of instances solved to optimality, for each combination of relative amplitude and offered load (for $\Delta_s = 60$)

column in the table contains the number of instances.

As indicated by the first column of Figure 5.8, the initial solution turns out to be optimal in 5.45% of the instances (0% cost reduction); verifying this may require a considerable number of simulations though. As shown in Table 5.5, the probability that the initial solution is optimal increases as (1) the service rate increases, (2) the offered load decreases, (3) the abandonment rate increases, (4) the waiting time target is less stringent, (5) the service and abandonment processes are less variable, and (6) the staffing intervals are large. This is not surprising, as all these factors limit the solution space (both the number of staffing intervals and the capacity required), so it can be expected that the heuristic solution is more likely to coincide with the estimated optimum.

Overall, a majority of instances (approx. 70%) could be solved to optimality quickly (less than 100 simulation runs, see Figure 5.8). The improvement with respect to the initial solution turned out to be relatively small in most cases (improvements of more than 5% were only observed in about 35.70% of the instances, with improvements exceeding 15% in only 2.47% of the instances). This mainly confirms that the quality of the initial solution, as generated by the ISA(τ) algorithm [65], is high (the related c_w^{init} tends to be close to the optimal shift cost). As confirmed by Table 5.5, cost improvements exceeding 5% were especially likely in settings with low service rates, or high variability in the service and abandonment processes.

Table 5.5 also provides more general insights into the optimal shift cost, across the different parameter settings. As expected, c_w^* increases as the offered load increases, and the relative amplitude of the arrival process increases. This is intuitive, as both factors imply that more capacity will be needed to meet the customer service constraint. Abandonments, by contrast, reduce the load on the system, and thus have a beneficial impact on the optimal shift cost. Furthermore, the staffing interval length plays a role: short staffing intervals provide more flexibility to the shift schedule, which tends to lead to lower optimal costs.

Finally, as observed before, the computational effort (as measured by the number of simulations performed) is highly sensitive to the size of the solution space, with the staffing interval length having a particularly large impact.

5.5.3 Impact of the number of replications

Any inaccuracies in the estimated customer service may affect the solution that is returned by the algorithm. In particular in steps 3 and 4 of Figure 5.5, nodes are fathomed based on the service level estimates, so inaccurate estimates may cause the algorithm to settle at a wrong optimum. As we use simulation to evaluate customer service, the estimation accuracy is impacted by the number of replications R. In this section, we compare $R = 100$ versus $R = 2500$, and assess the extent to which the difference in accuracy affects the computational effort required to run the algorithm to completion, and the observed cost difference at the final solution, for those instances that

Parameter	Values	Average optimal cost c_w^*	Average number of simulations (tree exploration)	Average proportion of instances where the initial solution is optimal	Average proportion of instances with cost reduction $> 5\%$	Number of instances
Service rate (μ)	1	162.333	554.462	0.000	0.831	249
	2	166.064	315.137	0.016	0.353	249
	4	159.108	72.281	0.161	0.169	249
Offered load (λ/μ)	5	96.559	33.751	0.110	0.486	245
	10	168.931	157.376	0.065	0.416	245
	15	239.078	455.482	0.033	0.327	245
Relative amplitude (RA)	0.5	121.040	144.007	0.073	0.442	392
	1	137.248	728.343	0.091	0.464	392
Abandonment rate (θ)	0	170.119	550.000	0.026	0.537	390
	μ	151.343	331.750	0.112	0.269	390
Maximum acceptable wait (τ)	0	198.754	124.873	0.056	0.349	252
	10	152.111	141.131	0.063	0.397	252
	20	134.147	566.071	0.091	0.548	252
C^2 of service and abandonment process (C^2)	0.5	169.629	212.367	0.110	0.295	264
	1	162.663	372.201	0.061	0.383	264
	2	153.902	400.530	0.030	0.595	264
Staffing interval length (Δ_s)	60	128.766	2715.509	0.070	0.287	171
	120	141.029	34.333	0.099	0.298	171
	240	143.766	22.094	0.135	0.374	171

Table 5.5: Sensitivity to system parameters ($R = 2500$), for instances solved to optimality

were solved to optimality.

More specifically, the difference in computational effort is measured through the number of simulation runs required (during the preprocessing and tree exploration stages):

$$\mathbf{\Delta}\text{SIM} \equiv \text{SIM}(R = 100) - \text{SIM}(R = 2500). \tag{5.8}$$

The cost difference (in percent) is determined as:

$$\mathbf{\Delta}c_w^* \equiv \frac{c_w^*(R = 100) - c_w^*(R = 2500)}{c_w^*(R = 2500)}. \tag{5.9}$$

Figure 5.9 shows that the difference in computational effort varies widely (with 5% and 95% percentiles equal to -662.4 and 251.4 respectively). We measured CPU times[2] of about 4 minutes on average for $R = 100$ (with 5% and 95% percentiles equal to 0.006 and 24.6 minutes respectively); for $R = 2500$ the average was about 60 minutes (with 5% and 95% percentiles equal to 0.1 and 377.5 minutes respectively).

The difference in cost, by contrast, is far less outspoken: increasing the number of replications has only a limited impact on the cost of the final solution (see Figure 5.10). Using $R = 100$ yields a c_w^* that is 1.84% higher on average (the 5% and 95% percentiles equal -1.69% and 7.14% respectively). The lower accuracy for R equal to 100 gives rise to performance estimates that tend to be more "noisy", compared to $R = 2500$. Because we impose a strict bound on performance, such inaccuracies may cause overstaffing (for $R = 100$). This explains the asymmetric distribution that can be observed in Figure 5.10.

The time-average of the confidence interval halfwidth around $\Pr(W_t > \tau)$ is about 4% on average (for $R = 100$) and 0.8% (for $R = 2500$). We found that in 32.6% of the instances, the final solutions obtained with $R = 100$ appear to be infeasible if they are evaluated with $R = 2500$. Though the performance constraint was typically violated in only a limited number of performance intervals, this shows that R should be large in settings where the performance constraint is strict.

[2]All experiments were performed with general model settings, but adjusting the parameters in the simulation model (such as maximum array sizes) more to the problem setting

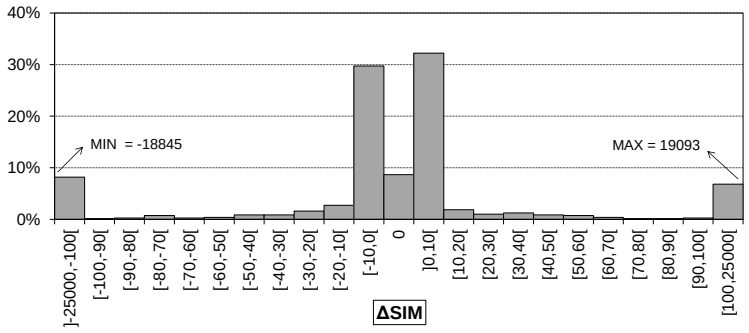


Figure 5.9: Sensitivity of the optimal solution to number of replications: difference in number of simulation runs

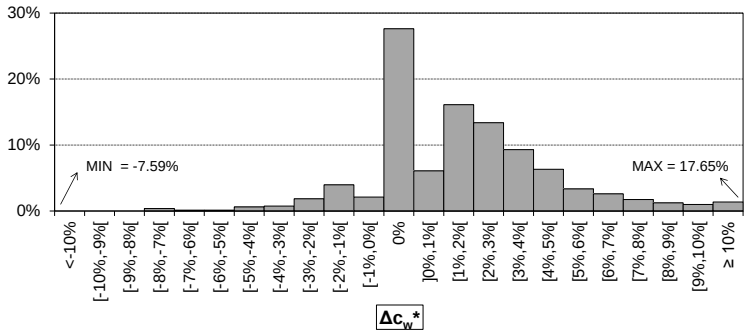


Figure 5.10: Sensitivity of the optimal solution to number of replications: difference in optimal shift cost

5.5.4 Impact of the initial solution

In all computational results shown so far, the initial solution was generated by the ISA(τ) algorithm [65]. As evident from the results, this initial solution tends to be of high quality. In this section, we explore how a lower-quality initial solution affects the number of simulations required to terminate the algorithm, and the speed with which an estimated optimal solution is found. Because the algorithm is stopped after a fixed number of simulated nodes, it is important that good feasible solutions are found quickly, even if their

is likely to reduce the CPU time notably. As such, these figures should be considered as a rough indication of the computation time.

optimality is not guaranteed. Ideally, the algorithm's speed in finding the optimal solution should not be impacted too severely by the quality of the initial solution.

The purpose of the initial feasible solution is twofold: it enables using the fathoming rules defined in Section 5.4.2 (it provides a value for c_w^{init}), and speeds up the search for the lower bound on the staffing requirements (which defines the root node of the tree). In this section, we apply an alternative initial solution that is simpler to calculate (it requires no simulation runs) but results in a higher initial shift cost. More specifically, $\mathbf{s}^{\mathsf{init}}$ is obtained as the smallest staffing vector that satisfies the delay probability constraint (i.e., τ equal to 0) in a stationary $M/M/s$ model with arrival rate $\lambda_{\mathsf{max}} = \max\{\lambda_t : t \in [0, T]\}$. This vector is feasible in the corresponding $M_t/M/s_t + M$ model (although it is probably very costly). In our experiments, the feasibility remains valid for general service and abandonment times, due to the large amount of excess capacity that is added due to the overly restrictive assumptions that are used (i.e., no abandonments, the use of λ_{max} and τ equal to 0).

Figure 5.11 contains the difference in the total number of simulation runs in the algorithm, for the instances that were solved to optimality (the alternative initial solution is indicated by $M/M/s$). The figure reveals that the total of simulations tends to be lower for ISA(τ). A paired t-test showed that the difference is significant (with $p < 0.01$). As such, the simulation runs required to determine the ISA(τ) solution result in a more than proportional reduction of the number of simulations needed to run the algorithm to completion. The algorithm succeeds in finding good solutions quickly, irrespective of the start solution: the differences in total simulation effort are generally small, even though the initial staffing cost c_s^{init} corresponding to the $M/M/s$-based solution may be substantially higher. Indeed, Table 5.6 shows that the best solution is typically found after a low number of simulation runs for both initial solutions (though the algorithm may require a substantial number of simulation runs to terminate).

Figure 5.12 provides further details on how the difference in initial staffing cost affects the algorithm's performance. It shows the percentage of test instances (solved to optimality), as a function of the difference in initial

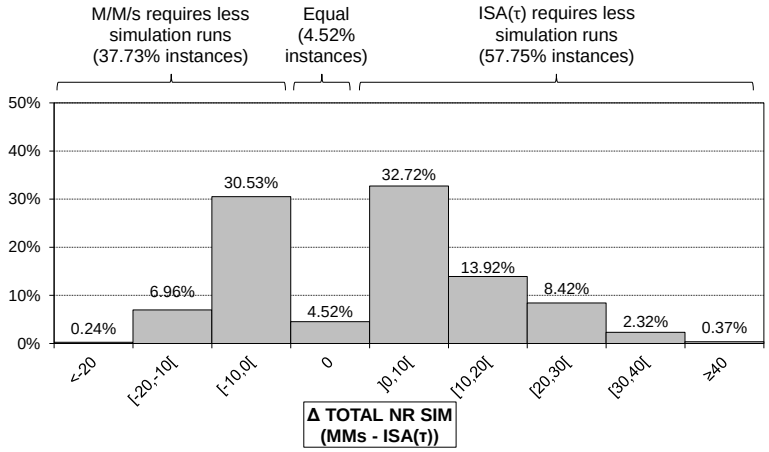


Figure 5.11: Difference in total number of simulations ($M/M/s$ - ISA(τ))

	Min	5% percentile	Median	95% percentile	Max	Average
ISA(τ)	11	18	35	2977	22671	606.04
$M/M/s$	16	18	39	2997	22686	609.68

Table 5.6: Comparison: total number of simulations required to reach an estimated optimum.

staffing cost and the total number of simulations. The figure reveals that the $M/M/s$-based solution outperforms the ISA(τ) solution only if its staffing cost is close to that of the ISA(τ) solution (note that this information is not available in advance). In that case, determining the ISA(τ) solution is not worthwhile the additional simulation effort. The staffing costs, however, often differ greatly: in 17% of instances, $c_s^{\text{init}}(M/M/s)$ is more than twice as large as $c_s^{\text{init}}(\text{ISA}(\tau))$. In those settings, the ISA(τ) solution clearly outperforms the $M/M/s$-based solution.

5.6 Conclusions and future research

We present an implicit enumeration approach to estimate optimal shift schedules in terminating systems with nonstationary arrivals and service

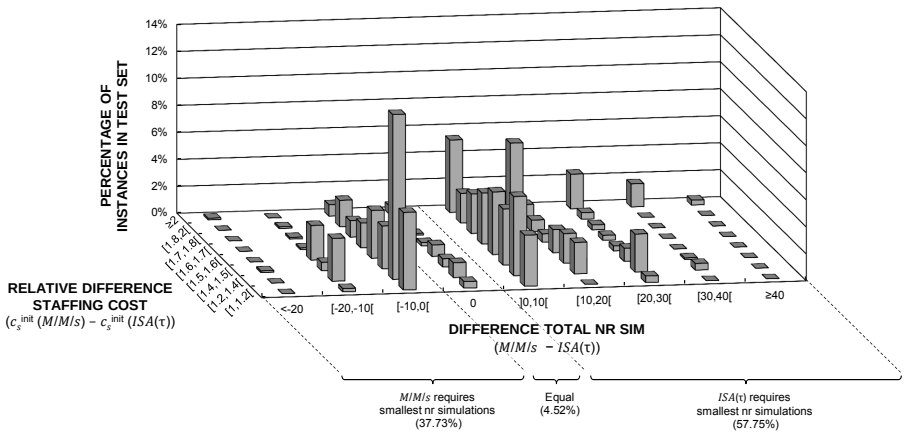


Figure 5.12: Impact of the initial solution: classification of test instances based on difference in number of simulations and difference in initial staffing cost

level constraints. The results show that the algorithm is efficient in exploring the solution space, though the computational effort increases significantly as the number of staffing intervals and the server requirements per interval increase. Consequently, the algorithm is best suited for small-scale systems, with a limited number of operators.

The algorithm is efficient and an estimated optimum is typically found quickly (even if an inferior start solution is used). The algorithm does not depend on a particular methodology to evaluate the service level constraints; in principle, any type of methodology can be used. However, the quality of the optimal solution proposed by the algorithm evidently depends on the *accuracy* of the customer service estimates.

The optimal solution found by our method is an *estimated* optimum because discrete-event simulation is used to estimate the service levels, and because alternative optima for Problem (5.2-5.4) are not accounted for. As is discussed in Section 5.4.2.2, the existence of alternative optima could cause the algorithm to miss the optimum in settings with an exhaustive service policy. This limitation especially holds for highly utilized systems with long service times, because the exhaustive service policy is most prominent in such settings. Though our approach cannot strictly guarantee the optimum

in the exhaustive setting, it will converge to the optimal solution in systems with a preemptive service policy (where service can be interrupted and the customer in service rejoins the queue), as the number of replications grows to infinity. Our approach can be extended to more realistic problems by replacing Problem (5.2-5.4) by dedicated scheduling or rostering algorithms. However, as Problem (5.2-5.4) may be solved many times in our algorithm, the efficiency of these algorithms will be a key determinant of the total computation time.

Two additional limitations of our model follow from the choice of the objective function. First, server overtime is not included as a cost component in the objective. The amount of overtime that follows from a schedule can be significant (especially if service times are long) and may need to be monitored. Second, we expressed the shift cost in man-hours. It is known that particular shift types tend to be more expensive (e.g., night or weekend shifts), yet, our model only differentiates between shifts based on the number of man-hours they require.

In future research, we plan to use our method to evaluate the solution quality of heuristic approaches available in the literature (such as [120, 122], among others) and to extend the approach towards nonterminating settings.

Chapter 6

Epilogue

Nonstationary arrival patterns, where the customer arrival rate fluctuates over the course of a day, can be observed in many service organizations (e.g., emergency departments, call centers, banks, and retail stores). In this dissertation, we have studied how personnel capacity planning can be used to control customer waiting times in the presence of time-varying demand for service. Our main contributions —from an operational point of view— can be found in Chapters 3 to 5.

First, we have looked into the issue of performance measurement in systems with nonstationary demand (i.e., how the time-varying waiting times can be quantified). Our experiments in Chapter 3 provide insights that are relevant for practitioners that are active in workforce planning: we found that the use of discrete-event simulation models consistently delivers good results and that simulation therefore provides a viable way to estimate customer waiting times in systems with nonstationary demand. The speed-accuracy trade-off of simulation is especially favorable in settings that do not comply with theoretical assumptions that are common in the academic literature (such as Poisson assumptions for service and abandonment processes). As such, simulation can successfully support the capacity planning process in this environment.

Next, we have put forward two methods for staffing (Chapter 4) and shift scheduling (Chapter 5). Our experiments in Chapter 4 revealed that the service policy (exhaustive versus preemptive) has a notable impact on the

system performance. Moreover, the exhaustive policy can lead to counterintuitive effects in staffing problems with short staffing intervals, in particular when the use of overtime is not penalized in the objective function (which –surprisingly– is common practice in the academic literature on nonstationary arrival processes). The latter issue is particularly problematic when the resulting staffing levels are used as input in a shift scheduling algorithm. To the best of our knowledge, our experiments are the first to highlight this impact. Developing methods that account for the overtime cost and that fully acknowledge the impact of the service policy in the context of time-varying arrival rates, are yet to be developed.

The scheduling algorithm presented in Chapter 5 exhibits some limitations, that may require further research. Firstly, our method is intended for small-scale systems with limited opening hours. Secondly, we used man-hours as a proxy of schedule cost. Man-hours, however, do not allow to make the full trade-off of labor cost and service (e.g., night shifts, weekend shifts and overtime are typically more expensive). Likewise, the use of atypical shift structures may induce different costs for the organization. These considerations may affect the choice of a personnel schedule and may call for continued research. On a final note, we remark that although we constructed a shift schedule based on a diverse set of shift types, we have not elaborated on the practical point of view, i.e., *how* shift flexibility can be realized within service organizations. As discussed in Chapter 1, the use part-time labor or so-called split shifts may provide organizations with higher flexibility for shift scheduling.

In spite of the contributions of this thesis, many challenges remain for continued research. Taking a broader perspective to the problem, we put forward the following main issues, which seem underexposed in current research:

- **Managing the uncertainties.** Gans et al. [85] distinguish between several types of uncertainty: *process uncertainty* captures the inherent randomness of the problem. *Parameter uncertainty* expresses uncertainty in the estimates of the distributional parameters (see Chapter 2). *Model uncertainty* relates to modeling assumptions, that cause the model to be merely an approximation of reality (e.g., assuming an

exponential distribution for the service process may be too stringent).

Our research has shown that model uncertainty should be monitored: the experiments in Chapters 3 and 4 indicate that although stationary approximations are appealing due to their simplicity, they are often too crude to yield consistently good outcomes. Similarly, we found that that the assumption regarding the service policy matters (see Chapter 3), and that approximating general distributions by the commonly used exponential distribution is not advisable (see Chapter 4). Though our research accounts for process uncertainty and fair degree of model uncertainty (time-dependent arrival rates, general distributions for the service process, customer abandonments, among others), our work is limited in the sense that we did not include parameter uncertainty. The development of robust optimization models that are capable of handling different types of uncertainty provides a promising direction for continued research and is increasing in popularity in the academic literature (see e.g., [252, 105, 28, 29]).

- **Moving toward integrated capacity planning.** The suboptimality that is introduced by decoupling the different steps of the personnel planning process (demand forecasting, staffing, shift scheduling, rostering), has motivated us to develop the integrated approach to staffing and shift scheduling (in Chapter 5). Similarly, it should be explored how demand forecasting and rostering decisions can efficiently be integrated with staffing and scheduling decisions, in models with nonstationary demand for service (though the problem complexity increases severely). Some efforts have been made in this direction: e.g., Gans et al. [86] introduced an integrated approach for forecasting, staffing and scheduling with time-varying demand and parameter uncertainty. Integrated approaches for staffing and rostering exist in airline crew scheduling [27] and nurse rostering [151] (though these apply to deterministic time-varying arrivals).

  Furthermore, we emphasize that time-dependent arrivals are equally relevant for determining the capacity of other resources (material, infrastructure, and other resources). For instance, Stolletz [224] analyze

runway capacity at airports with time-dependent demand, and Zhang et al. [259] develop an approach for long-term bed capacity planning, that is inspired by the heuristics from the call center literature. This shows that synergies may exist between the models developed for personnel and those targeting other resource types.

Lastly, we wish to re-emphasize that capacity planning is not the only approach for coping with time-varying demand: depending upon the actual context, various demand management strategies can be applied to influence the arrival rates, such that they are more in line with the available capacity (as discussed in Chapter 1). A joint strategy that includes both capacity and demand management is advisable [147, 119, 215, 147]. This can be done by, for instance, first manipulating the arrival rate and then adjusting capacity to the fluctuations in smoothed demand (cf. the "Influence" strategy of Crandall and Markland [56], see Chapter 1). Achieving alignment between demand management and capacity management (for personnel and other resources) poses a broad-scope challenge for academics.

Appendix A

Infinite server offered load for hypo-exponential and two-phase Coxian distributions

Let λ_t represent the time-varying arrival rate function, such that it follows a sine function with parameters a, b, and c. Formally,

$$\lambda_t = a + b \sin\left(\frac{2\pi t}{c}\right),$$

with a equal to the time-average of λ_t, b indicating the amplitude of the sine, and c representing the period. The infinite server offered load at a given time t equals [99, 78]:

$$\begin{aligned} m_t^{\infty} &= \int_{-\infty}^{t} \left[1 - G_{t-u}\right] \lambda_u \mathrm{d}u \\ &= \int_{-\infty}^{t} \left[1 - \int_{0}^{t-u} f_x \mathrm{d}x\right] \lambda_u \mathrm{d}u\,, \end{aligned}$$

where $G_{(\cdot)}$ is the cumulative distribution of the service process, and $f_{(\cdot)}$ indicates the corresponding PDF. The service process is characterized by rate μ and the squared coefficient of variation C^2.

The service process is modeled as a hypo-exponential, two-phase Coxian or exponential distribution, depending on the value of C^2. For further details, see Creemers et al. [57]. For $C^2 < 1$, the service distribution is approximated by a hypo-exponential distribution, consisting of Z exponential phases with rate μ_1, followed by a single exponential phase with rate μ_2. The number of phases Z is equal to [57]:

$$Z = \lceil C^{-2} \rceil .$$

The parameters μ_1 and μ_2 can be obtained from

$$\mu_1 = \frac{(Z-1) - \sqrt{(Z-1)\left(ZC^2-1\right)}}{(1/\mu)\left(1-C^2\right)} \text{ , and}$$
$$\mu_2 = \frac{1 + \sqrt{(Z-1)\left(ZC^2-1\right)}}{(1/\mu)\left(1-ZC^2+C^2\right)} .$$

In our experiments, C^2 equals 0.5, so Z equals 2. As a result, μ_1 equals μ_2, and as such, the service distribution is equivalent to an Erlang distribution of two phases. The corresponding PDF, denoted $f^{\text{erl2}}_{(\cdot)}$, can be obtained from:

$$f^{\text{erl2}}_x = \mu_1^2 x e^{-\mu_1 x} .$$

A closed-form expression for the infinite server offered load can be derived for $C^2 = 0.5$:

$$\begin{aligned} m_t^\infty &= \int_{-\infty}^{t} \left[1 - \int_0^{t-u} f^{\text{erl2}}_x \mathrm{d}x\right] (a + b\sin(2\pi u/c))\,\mathrm{d}u \\ &= \frac{2\left(a((c\mu_1)^2+4\pi^2)^2 - bc\mu_1\pi(3(c\mu_1)^2+4\pi^2)\cos(\frac{2\pi t}{c}) + bc^4\mu_1^4\sin(\frac{2\pi t}{c})\right)}{\mu_1((c\mu_1)^2+4\pi^2)^2} \end{aligned} \tag{A.1}$$

For $C^2 > 1$, a two-phase Coxian distribution is used, consisting of a single exponential phase with rate μ_1, followed by a second phase with rate μ_2; the latter will be performed with probability β. The total service rate can be decomposed by a scaling factor κ (we used $\kappa = 0.5$, as in Creemers et al. [57]):

$$\mu_1 = \frac{\mu}{\kappa} .$$

The parameters μ_2 and β then can be obtained in terms of the parameters μ, C^2, and κ:

$$\mu_2 = \frac{2\,(\kappa - 1)}{(1/\mu)\,(2\kappa - 1 - C^2)} \text{ and}$$
$$\beta = \frac{2\,(\kappa - 1)^2}{1 + C^2 - 2\kappa}\,.$$

Because both phases are exponentially distributed, their PDFs can be expressed as:

$$f_x^{\text{expo1}} = \mu_1 e^{-\mu_1 x} \text{ , and}$$
$$f_x^{\text{expo2}} = \mu_2 e^{-\mu_2 x}\,.$$

Both phases are executed with probability β. The service process consists of only the first phase, with probability $(1 - \beta)$. The infinite server offered load for service times that follow a two-phase Coxian distribution, can then be expressed as:

$$\begin{aligned} m_t^\infty &= \int_{-\infty}^{t} \left[\beta \left(1 - \int_0^{t-u} \int_0^{t-u-x} f_x^{\text{expo1}} f_y^{\text{expo2}} \mathrm{d}y \mathrm{d}x \right) \right. \\ &\quad \left. + (1-\beta) \left(1 - \int_0^{t-u} f_x^{\text{expo1}} \mathrm{d}x \right) \right] \left(a + b \sin\left(\frac{2\pi u}{c} \right) \right) \mathrm{d}u \\ &= a \left(\frac{1}{\mu_1} + \frac{\beta}{\mu_2} \right) \\ &\quad + \frac{2bc\pi \sin t\mu_1 \left(-4\pi^2(-1+\beta) + c^2\mu_2(\beta\mu_1 + \mu_2) \right)}{\left(4\pi^2 + c^2\mu_1^2\right)\left(4\pi^2 + c^2\mu_2^2\right)} \\ &\quad - \frac{4b \cos \pi^2 t \left(4\pi^2 + c^2 \left(\mu_2^2 + \beta\mu_1(\mu_1 + \mu_2) \right) \right)}{\left(4\pi^2 + c^2\mu_1^2\right)\left(4\pi^2 + c^2\mu_2^2\right)}\,. \end{aligned} \tag{A.2}$$

In the computational experiment described in Section 5.5, Expressions (A.1) and (A.2) serve to obtain the infinite server offered load, as needed for MOL.

Appendix B

Relation between SIM-OAM and SIM-OWM

The performance metrics obtained through SIM-OAM and SIM-OWM are linked by the following relation:

$$E\left[\frac{L_{i_p}}{A_{i_p}}\right] = \frac{E\left[L_{i_p}\right]}{E\left[A_{i_p}\right]} - \frac{1}{E\left[A_{i_p}\right]} Cov\left(A_{i_p}, \frac{L_{i_p}}{A_{i_p}}\right). \tag{B.1}$$

We make a distinction between (1) settings where capacity remains constant and (2) settings where capacity is time-varying. If capacity remains constant, A_{i_p} and $\frac{L_{i_p}}{A_{i_p}}$ display a non-negative correlation: the number of customers with an excessive wait will never decrease if the number of arrivals in a given interval increases. Moreover, the correlation is likely to increase as the threshold value for an excessive wait, τ, decreases. For $\tau \to \infty$, it holds that $L_{i_p} \to 0$, for any given A_{i_p}. Thus, if the capacity remains constant, it holds that:

$$E\left[\frac{L(i_p)}{A(i_p)}\right] \leq \frac{E[L(i_p)]}{E[A(i_p)]}, \tag{B.2}$$

as was also reported in Maman [173].

In theory, $Cov\left(A_{i_p}, \frac{L_{i_p}}{A_{i_p}}\right)$ may be negative in some (exceptional) cases if the capacity can fluctuate over time (then, Expression B.2 does not hold). A simplistic example of such a setting is plotted in Figure B.1: although the number of arrivals is larger in the second replication, the ratio $\frac{L(i_p)}{A(i_p)}$

decreases due to the capacity increase. This may particularly occur in performance intervals that precede a capacity change and if τ is large (the effect disappears if $\tau = 0$).

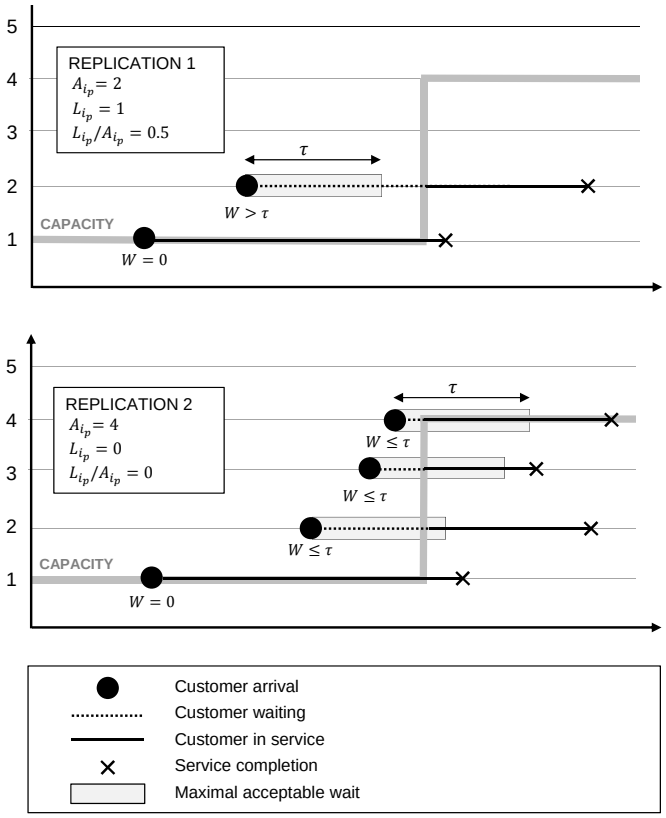


Figure B.1: Example for which $Cov\left(A_{i_p}, \frac{L_{i_p}}{A_{i_p}}\right) = -0.5 < 0$

Appendix C

Impact of scaling factor on algorithm convergence

	Small system				Large system			
	$i/2$	i	$2i$	i^2	$i/2$	i	$2i$	i^2
Number iterations Phase I	10	6	4	6	17	13	15	12
Number iterations Phase II	5	2	1	2	154	5	3	15
Total number simulations	15	8	5	8	171	18	18	27
$\max_{t \in \mathbf{t}_p}\{P_t\}$	0.095	0.090	0.090	0.090	0.100	0.099	0.099	0.100
Staffing cost c_s^*	72.50	74.25	74.25	74.00	2343.25	2296.00	2287.75	2394.25

Appendix D

Rescaling of α (finite vs. infinite patience)

To allow for a fair comparison with ISA(τ), the value of α needs to be rescaled for the LAGSIPP_SRS and MOL_SRS heuristics. Recall that we focus on virtual waiting times in the ISA(τ) approach (i.e., the waiting time of a fictive customer with *infinite* patience), hence α represents the target for the virtual probability of excessive waiting. LAGSIPP_SRS and MOL_SRS, however, are based on the waiting time of a customer with *finite* patience, which implies that waiting times may be lower due to customer abandonment. Let W_t^i, W_t^f and L represent the infinite patience waiting time, the finite patience waiting time and the time-to-abandon random variable. The relation between the infinite patience probability of excessive waiting and the finite patience probability of excessive waiting can be expressed as follows:

$$\begin{aligned}
\Pr(W_t^f > \tau) &= 1 - \Pr(W_t^f \leq \tau) && \text{(D.1)}\\
&= 1 - \Pr(\min(W_t^i, L) \leq \tau) && \text{(D.2)}\\
&= 1 - \Pr(W_t^i \leq \tau \cup L \leq \tau) && \text{(D.3)}\\
&= 1 - \big(\Pr(W_t^i \leq \tau) + \Pr(L \leq \tau) - \Pr(W_t^i \leq \tau)\Pr(L \leq \tau)\big) && \text{(D.4)}
\end{aligned}$$

Or, alternatively,

$$\Pr(W_t^i > \tau) = \frac{\Pr(W_t^f \leq \tau)}{1 - \Pr(L \leq \tau)}\,. \tag{D.5}$$

As such, a target α for $\Pr(W_t^i > \tau)$ corresponds to $\alpha' = \alpha(1 - \Pr(L \leq \tau))$ for $\Pr(W_t^f > \tau)$.

Though Expression D.1 remains valid for general distributions, we assume exponentially distributed abandonments, to comply with the $M/M/s+M$ assumption that is made by using the Garnett delay function in the LAGSIPP_SRS and MOL_SRS heuristics. In our computational experiment, only LAGSIPP_SRS and MOL_SRS require this rescaling. Although LAGSIPP_CF and MOL_CF also assume finite patience, rescaling α is not required: W_t^f and W_t^i are equal because abandonments do not occur (the $M_t/G/s_t + G$ queue is approximated by a series of $M/M/s$ queues). LAGSIPP_SIM and MOL_SIM are determined by a simulation model that uses virtual waiting times.

Appendix E

Bounds

Lower bound (LB). The LB is determined as follows: in each staffing interval i_s, $s_{i_s}^{\mathsf{LB}}$ is set equal to the smallest capacity level that is needed to meet the performance constraint, assuming that infinite capacity is available in all other staffing intervals. Ingolfsson et al. [122] suggest a similar approach, but start from an empty system. While Ingolfsson et al. [122] use bisection search to obtain $s_{i_s}^{\mathsf{LB}}$ for each i_s, we opt to make unit-size decreases starting from the feasible (heuristic) solution $s_{i_s}^{\mathsf{init}}$ obtained through ISA(τ)); as $s_{i_s}^{\mathsf{init}}$ tends to be relatively tight, we found that this approach finds the lower bound with fewer evaluations than bisection search. Note that, for $\tau \geq \Delta_s$, it suffices to set s^{LB} equal to 1 (assuming that at least 1 server should be available at all times): as τ spans multiple staffing intervals, the capacity shortage in any given interval is compensated by the infinite capacity in the following interval.

Upper bound (UB). The vector $\mathbf{s}^{\mathsf{UB}}$ contains an upper bound on the staffing requirement in each staffing interval. It is constructed based on the initial shift cost c_w^{init}. For each interval i_s, the cheapest shift that can be active in that interval is selected. Let this shift be represented by j_{min}, with shift cost $c_a^{j_{\mathsf{min}}}$. The upper bound in interval i_s is then determined as the largest number of shifts j that can be active that yields a total staffing cost of at most c_w^{init}:

$$s_{i_s}^{\mathsf{UB}} = \left\lfloor \frac{c_w^{\mathsf{init}}}{c_a^{j_{\mathsf{min}}}} \right\rfloor, \forall i_s \in \mathbf{I}_s \, .$$

All solutions for which $s_{i_s} > s_{i_s}^{\mathsf{UB}}$ in at least one staffing interval yield a staffing cost that exceeds c_w^{init}, and should not be considered in the search tree.

Appendix F

Shift specifications

Staffing interval length (number of shifts)	Shift specification {start time, end time, start time break}
$\Delta_s = 240$ ($W = 5$)	$\{0,4,-\}$, $\{4,8,-\}$, $\{8,12,-\}$, $\{0,8,-\}$, $\{4,12,-\}$
$\Delta_s = 120$ ($W = 12$)	$\{0,4,-\}$, $\{2,6,-\}$, $\{4,8,-\}$, $\{6,10,-\}$, $\{8,12,-\}$, $\{0,6,-\}$, $\{2,8,-\}$, $\{4,10,-\}$, $\{6,12,-\}$, $\{0,8,-\}$, $\{2,10,-\}$, $\{4,12,-\}$
$\Delta_s = 60$ ($W = 45$)	$\{0,4,-\}$, $\{1,5,-\}$, $\{2,6,-\}$, $\{3,7,-\}$, $\{4,8,-\}$, $\{5,9,-\}$, $\{6,10,-\}$, $\{7,11,-\}$, $\{8,12,-\}$, $\{0,6,2\}$, $\{1,7,3\}$, $\{2,8,4\}$, $\{3,9,5\}$, $\{4,10,6\}$, $\{5,11,7\}$, $\{6,12,8\}$, $\{0,6,3\}$, $\{1,7,4\}$, $\{2,8,5\}$, $\{3,9,6\}$, $\{4,10,7\}$, $\{5,11,8\}$, $\{6,12,9\}$, $\{0,6,4\}$, $\{1,7,5\}$, $\{2,8,6\}$, $\{3,9,7\}$, $\{4,10,8\}$, $\{5,11,9\}$, $\{6,12,10\}$, $\{0,8,3\}$, $\{1,9,4\}$, $\{2,10,5\}$, $\{3,11,6\}$, $\{4,12,7\}$, $\{0,8,4\}$, $\{1,9,5\}$, $\{2,10,6\}$, $\{3,11,7\}$, $\{4,12,8\}$, $\{0,8,5\}$, $\{1,9,6\}$, $\{2,10,7\}$, $\{3,11,8\}$, $\{4,12,9\}$

Table F.1: Shift specifications (all breaks are assumed to be 1 hour). W represents the size of the shift set for problem instances with staffing interval length Δ_s

List of Figures

List of Tables

Bibliography

[1] Adenso-Diaz, B., P. González-Torre, V. Garcia. 2002. A capacity management model in service industries. *International Journal of Service Industry Management* **13**(3) 286–302.

[2] Agnihothri, S. R., P. F. Taylor. 1991. Staffing a centralized appointment scheduling department in Lourdes hospital. *Interfaces* **21**(5) 1–11.

[3] Aguir, S., F. Karaesmen, O. Z. Akskin, F. Chauvet. 2004. The impact of retrials on call center performance. *OR Spectrum* **26**(3) 353–376.

[4] Aguir, S., O. Z. Aksin, F. Karaesmen, Y. Dallery. 2008. On the interaction between retrials and sizing of call centers. *European Journal of Operational Research* **191**(2) 398–408.

[5] Ahmed, M. A., T. M. Alkhamis. 2009. Simulation optimization for an emergency department healthcare unit in Kuwait. *European Journal of Operational Research* **198**(3) 936–942.

[6] Akbari, M., M. Zandieh, B. Dorri, B. 2013. Scheduling part-time and mixed-skilled workers to maximize employee satisfaction. *The International Journal of Advanced Manufacturing Technology* **64**(5-8) 1017–1027.

[7] Aksin, Z., M. Armony, V. Mehrotra. 2007. The modern call center: A multi-disciplinary perspective on operations management research. *Production and Operations Management* **16**(6) 665–688.

[8] Aksin, O. Z., F. Karaesmen, E. L. Ormeci. 2007. A review of workforce cross-training in call centers from an operations management perspective. D. Nembhard (ed.), *Workforce Cross Training Handbook*. CRC Press, Boca Raton, USA.

[9] Aldor-Noiman, S., P. D. Feigin, A. Mandelbaum. 2009. Workload forcasting for a call center: methodology and a case study. *Annals of Applied Statistics* **3**(4) 1403–1447.

[10] Altman, E., T. Jiménez, G. Koole. 2001. On the comparison of queueing systems with their fluid limits. *Probability in the Engineering and Informational Sciences* **15**(2) 165–178.

[11] Andrews, B., H. Parsons. 1993. Establishing telephone-agent staffing levels through economic optimization. *Interfaces* **23**(2) 14–20.

[12] Armistead, C., G. Clark. 1994. The "coping" capacity management strategy in services and the influence on quality performance. *International Journal of Service Industry Management* **5**(2) 5–22.

[13] Atlason, J., M. A. Epelman, S. G. Henderson. 2004. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research* **127**(1) 333–358.

[14] Atlason, J., M. A. Epelman, S. G. Henderson. 2008. Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Science* **54**(2) 295–309.

[15] Avramidis, A. N., A. Deslauriers, P. L'Ecuyer. 2004. Modeling daily arrivals to a telephone call center. *Management Science* **50**(7) 896–908.

[16] Avramidis, A. N., W. Chan, P. L'Ecuyer. 2009. Staffing multi-skill call centers via search methods and a performance approximation. *IIE Transactions* **41**(6) 483–497.

[17] Avramidis, A. N., W. Chan, M. Gendreau, P. LEcuyer, O. Pisacane. 2010. Optimizing daily agent scheduling in a multiskill call center. *European Journal of Operational Research* **200**(3) 822–832.

[18] Aykin, T. 1996. Optimal shift scheduling with multiple break windows. *Management Science* **42**(4) 591–602.

[19] Aykin, T. 2000. A comparative evaluation of modeling approaches to the labor shift scheduling problem. *European Journal of Operational Research* **125**(2) 381–397.

[20] Baccelli, F., G. Hebuterne. 1981. On queues with impatient customers. F. J. Kylstra (ed.), *Performance 81*, North-Holland, Amsterdam, 159-179.

[21] Bard, J. F., H. W. Purnomo. 2006. Incremental changes in the workforce to accommodate changes in demand. *Health Care Management Science* **9**(1) 71–85.

[22] Baron, O., J. Milner. 2009. Staffing to maximize profit for call centers with alternate service-level agreements. *Operations Research* **57**(3) 685–700.

[23] Bassamboo, A., A. Zeevi. 2005. Dynamic routing and admission control in high-volume service systems: asymptotic analysis via multi-scale fluid limits. *Queueing Systems* **51**(3-4) 249–285.

[24] Bassamboo, A., J. M. Harrison, A. Zeevi. 2006. Design and control of a large call center: asymptotic analysis of an LP-based method. *Operations Research* **54**(3) 419–435.

[25] Bassamboo, A., A. Zeevi. 2009. On a data-driven method for staffing large call centers. *Operations Research* **57**(3) 714–726.

[26] Bechtold, S. E., L. W. Jacobs. 1990. Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science* **36**(11) 1339–1351.

[27] Beliën, J., E. Demeulemeester, P. De Bruecker, J. Van den Bergh, B. Cardoen. 2013. Integrated staffing and scheduling for an aircraft line maintenance problem. *Computers & Operations Research* **40**(4) 1023–1033.

[28] Bertsimas, D., X. V. Doan. 2010. Robust and data-driven approaches to call centers. *European Journal of Operational Research* **207**(2) 1072–1085.

[29] Bertsimas, D., V. Gupta, N. Kallus. 2013. Data-Driven Robust Optimization. *arXiv preprint*, arXiv:1401.0212.

[30] Betts, A., M. Meadows, P. Walley. 2000. Call center capacity management. *International Journal of Service Industry Management* **11**(2) 185–96.

[31] Bhandari, A., A. Scheller-Wolf, M. Harchol-Balter. 2008. An exact and efficient algorithm for the constrained dynamic operator staffing problem for call centers. *Management Science* **54**(2) 339–353.

[32] Bhulai, S., G. Koole, A. Pot. 2008. Simple methods for shift scheduling in multiskill call centers. *Manufacturing & Service Operations Management* **10**(3) 411–420.

[33] Boivin, D. B., G. M. Tremblay, F. O. James. 2007. Working on atypical schedules. *Sleep medicine* **8**(6) 578–589.

[34] Bolotin, V. A. 1994. Telephone circuit holding time distributions. J. Labetoulle and J. W. Roberts (eds.), *Proceedings of the 14th International Teletraffic Conference*, Elsevier, Amsterdam, Netherlands, 125–134.

[35] Borst, S., A. Mandelbaum, M. I. Reiman. 2004. Dimensioning large call centers. *Operations Research* **52**(1) 17–34.

[36] Brahimi, M., D. J. Worthington. 1991. The finite capacity multi-server queue with inhomogeneous arrival rate and discrete service time distribution and its application to continuous service time problems. *European Journal of Operational Research* **50**(3) 310–324.

[37] Brahimi, M. 1990. Approximating multi-server queues with inhomogeneous arrival rates and continuous service time distributions. *PhD Dissertation*, University of Lancaster, Lancaster, UK.

[38] Brown, L., N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, L. Zhao. 2005. Statistical analysis of a telephone call center: A queueing perspective. *Journal of the American Statistical Association* **100**(469) 36–50.

[39] Brunner, J. O., J. F. Bard, R. Kolisch. 2010. Midterm scheduling of physicians with flexible shifts using branch and price. *IIE Transactions* **43**(2) 84–109.

[40] Brusco, M. J., T. R. Johns. 1998. Staffing a multiskilled workforce with varying levels of productivity: An analysis of cross-training policies. *Decision Sciences* **29**(2) 499–515.

[41] Buesching, D. P., A. Jablonowski, E. Vesta, W. Dilts, C. Runge, J. Lund, R. Porter. 1985. Inappropriate emergency department visits. *Annals of Emergency Medicine* **14**(7) 672–676.

[42] Buffa, E. S., M. J. Cosgrove, B. J. Luce. 1976. An integrated work shift scheduling system. *Decision Sciences* **7**(4) 620–630.

[43] Burke, E. K., P. De Causmaecker, G. Vanden Berghe, H. Van Landeghem. 2004. The state of the art of nurse rostering. *Journal of Scheduling* **7**(1) 441–499.

[44] Campello, F., A. Ingolfsson. 2011. Exact necessary staffing requirements based on stochastic comparisons with infinite-server models. *Working paper*, University of Alberta, Canada.

[45] Carret, M. L., A. G. Fassa, I. Kawachi. 2007. Demand for emergency health service: factors associated with inappropriate use. *BMC Health Services Research* **7**(131).

[46] Castillo, I., T. Joro, Y.Y. Li. 2009. Workforce scheduling with multiple objectives. *European Journal of Operational Research* **196**(1) 162–170.

[47] Centeno, M. A., R. Giachetti, R. Linn, A. M. Ismail. 2003. Emergency departments II: a simulation-ILP based tool for scheduling ER staff. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice (eds.) *Proceedings of the 35th conference on Winter simulation: driving innovation* 1930–1938.

[48] Chase, R. B., U. M. Apte. 2007. A history of research in service operations: What's the big idea? *Journal of Operations Management* **25**(2) 375–386.

[49] Chassioti, E., D. J. Worthington. 2004. A new model for call centre queue management. *The Journal of the Operational Research Society* **55**(12) 1352–1357.

[50] Chassioti, E., D. Worthington, K. Glazebrook. 2013. Effects of state-dependent balking on multi-server non-stationary queueing systems. *Journal of the Operational Research Society* **1** 1–13.

[51] Chen, B. K. P, S. G. Henderson. 2001. Two issues in setting call centre staffing levels. *Annals of Operations Research* **108**(1-4) 175–192.

[52] Choi, K., J. Hwang, M. Park. 2009. Scheduling restaurant workers to minimize labor cost and meet service standards. *Cornell Hospitality Quarterly* **50**(2) 155–167.

[53] Choudhury, G. L., A. Mandelbaum, M. I. Reiman, W. Whitt. 1997. Fluid and diffusion limits for queues in slowly changing random environments. *Stochastic Models* **13**(1) 121–146.

[54] Clark, G. M. 1981. Use of Polya distributions in approximate solutions to nonstationary $M/M/s$ queues. *Commun. ACM* **24**(4) 206–217.

[55] Corominas, A., A. Lusa. 2012. LETRIS: Staffing service systems by means of simulation. *Journal of Industrial Engineering and Management* **5**(2) 285–296.

[56] Crandall, R. E, R. E Markland. 1996. Demand management – todays challenge for service industries. *Production and Operations Management* **5**(2) 106–120.

[57] Creemers, S., M. Defraeye, I. Van Nieuwenhuyse. 2013. A Markov model for measuring service levels in nonstationary $G_t/G_t/s_t + G_t$ queues. *Research report KBI_1306*, KU Leuven, Leuven, Belgium.

[58] Dacko, S. G. 2012. Time-of-day services marketing. *Journal of Services Marketing* **26**(5) 375–388.

[59] Dai, J. G., He, S. 2010. Customer abandonment in many-server queues. *Mathematics of Operations Research* **35**(2) 347–362.

[60] Dantzig, G. 1954. A comment on Edies traffic delay at toll booths. *Operations Research* **2** 339–341.

[61] Davis, J. L., W. A. Massey, W. Whitt. 1995. Sensitivity to the service-time distribution in the nonstationary erlang loss model. *Management Science* **41**(6) 1107–1116.

[62] De Causmaecker, P., G. Vanden Berghe. 2011. A categorisation of nurse rostering problems. *Journal of Scheduling* **14**(1) 3–16.

[63] Dean, A. M. 2002. Service quality in call centres: implications for customer loyalty. Managing Service Quality **12**(6) 414–423.

[64] Defraeye, M., I. Van Nieuwenhuyse. 2011. Setting staffing levels in an emergency department: opportunities and limitations of stationary queuing models. *Review of Business and Economics* **56**(1) 73–100.

[65] Defraeye, M., I. Van Nieuwenhuyse. 2013. Controlling excessive waiting times in small service systems with time-varying demand: An extension of the ISA algorithm. *Decision Support Systems* **54**(4) 1558–1567.

[66] Dembe, A. E., J. B. Erickson, R. G. Delbos, S. M. Banks. 2006. Nonstandard shift schedules and the risk of job-related injuries. *Scandinavian journal of work, environment & health* 232–240.

[67] Demers, S., A. Palmer, C. T. Griffiths. 2007. Vancouver Police Department Patrol Deployment Study. City of Vancouver.

[68] Dietz, D. C. 2011. Practical scheduling for call center operations. *Omega* **39** 550–557.

[69] Easton, F. F., D. F. Rossin. 1997. Overtime schedules for full-time service workers. *Omega* **25**(3) 285–299.

[70] Eick, S. G., W.A. Massey, W. Whitt. 1993a. The physics of the $M_t/G/\infty$ queue. *Operations Research* **41**(4) 731–742.

[71] Eick, S. G., W. A. Massey, W. Whitt. 1993b. $M_t/G/\infty$ queues with sinusoidal arrival rates. *Management Science* **39**(2) 241–252.

[72] Erdoğan, G., E. Erkut, A. Ingolfsson, G. Laporte. 2010. Scheduling ambulance crews for maximum coverage. *Journal of the Operational Research Society* **61**(4) 543–550.

[73] Ernst, A. T., H. Jiang, M. Krishnamoorthy, D. Sier. 2004. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research* **153**(1) 3–27.

[74] Ernst, A. T., H. Jiang, M. Krishnamoorthy, B. Owens, D. Sier. 2004. An Annotated Bibliography of Personnel Scheduling and Rostering. *Annals of Operations Research* **127**(1-4) 21–144.

[75] Ertogral, K., B. Bamuqabel. 2008. Developing staff schedules for a bilingual telecommunication call center with flexible workers.*Computers & Industrial Engineering* **54**(1) 118–127.

[76] Evans, G. W., T. B. Gor, E. Unger. 1996. A simulation model for evaluating personnel schedules in a hospital emergency department. J. M. Charnes, D. J. Morrice, D. T. Brunner, J. J. Swain (Eds.), *Proceedings of the 28th conference on Winter simulation (WSC '96)*, IEEE Computer Society, Washington, DC, USA, 1205–1209.

[77] Feldman, Z., A. Mandelbaum, W. A. Massey, W. Whitt. 2005. Staffing of time-varying queues to achieve time-stable performance. *Working paper.*

[78] Feldman, Z., A. Mandelbaum, W. A. Massey, W. Whitt. 2008. Staffing of time-varying queues to achieve time-stable performance. *Management Science* **54**(2) 324–338.

[79] Fitzsimmons, J. A. 1985. Consumer participation and productivity in service operations. *Interfaces* **15**(3) 60–67.

[80] Fletcher, A., D. Halsall, S. Huxham, D. Worthington. 2007. The DH accident and emergency department model: a national generic model used locally. *Journal of the Operational Research Society* **58** 1554–1562.

[81] Fletcher, A., D. J. Worthington. 2007. What is a "generic" hospital model? *Working Paper*, Department of Management Science, Lancaster University, Lancaster, UK.

[82] Fu, M. C., S. I. Marcus, I. J. Wang. 2000. Monotone optimal policies for a transient queueing staffing problem. *Operations Research* **48**(2) 327–331.

[83] Fukunaga, A., E. Hamilton, J. Fama, D. Andre, O. Matan, I. Nourbakhsh. 2002. Staff scheduling for inbound call centers and customer contact centers. *AI Magazine* **23**(4) 30–40.

[84] Gans, N., N. Liu, A. Mandelbaum, H. Shen, H. Ye. 2010. Service times in call centers: Agent heterogeneity and learning with some operational consequences. *Festschrift for Lawrence D. Brown* **6** of IMS Collections, Beachwood, 99–123.

[85] Gans, N., G. Koole, A. Mandelbaum. 2003. Telephone call centers: Tutorial, review, and research prospect. *Manufacturing & Service Operations Management* **5**(2) 79–141.

[86] Gans, N., H. Sheng, Y.-P. Zhou, N. Korolev, A. McCord, H. Ristock. 2012. Parametric stochastic programming models for call-center workforce scheduling. *Working paper*, University of Washington, Washington, USA.

[87] Garcia, M. L, M. A. Centeno, C. Rivera, N. DeCario. 1995. Reducing time in an emergency room via a fast-track. C. Alexopoulos, K. Kang (eds.), *Proceedings of the 27th conference on Winter simulation (WSC '95)*, IEEE Computer Society, Washington, DC, USA, 1048–1053.

[88] Garnett, O., A. Mandelbaum, M. Reiman. 2002. Designing a call center with impatient customers. *Manufacturing & Service Operations Management* **4**(3) 208–227.

[89] Gnanlet, A., W. G. Gilland. 2009. Sequential and simultaneous decision making for optimizing health care resource flexibilities. *Decision Sciences* **40**(2) 295–326.

[90] Grassmann, W. K. 1977. Transient solutions in Markovian queueing systems. *Computers & Operations Research* **4**(1) 47–53.

[91] Green, L. V., P. J. Kolesar. 1991. The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science* **37**(1) 84–97.

[92] Green, L. V., P. J. Kolesar. 1995. On the accuracy of the simple peak hour approximation for markovian queues. *Management Science* **41**(8) 1353–1370.

[93] Green, L. V., P. J. Kolesar. 1997. The Lagged PSA for Estimating Peak Congestion in Multiserver Markovian Queues with Periodic Arrival Rates. *Management Science* **43**(1) 80–87.

[94] Green, L. V., J. Soares. 2007. Computing time-dependent waiting time probabilities in $M(t)/M/s(t)$ queueing systems. *Manufacturing & service operations management* **9**(1) 54–61.

[95] Green, L. V., P. J. Kolesar, A. Svoronos. 1991. Some effects of nonstationarity on multiserver markovian queueing systems. *Operations Research* **39**(3) 502–511.

[96] Green, L. V., P. J. Kolesar, J. Soares. 2001. Improving the SIPP approach for staffing service systems that have cyclic demands. *Operations Research* **49**(4) 549–564.

[97] Green, L. V., P. J. Kolesar, J. Soares. 2003. An improved heuristic for staffing telephone call centers with limited operating hours. *Production and Operations Management* **12**(1) 46–61.

[98] Green, L. V., J. Soares, J. F. Giglio, R. A. Green. 2006. Using queueing theory to increase the effectiveness of emergency department provider staffing. *Academic Emergency Medicine* **13**(1) 61–68.

[99] Green, L. V., P. J. Kolesar, W. Whitt. 2007. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management* **16**(1) 13–39.

[100] Green, L. V. 2005. Capacity planning and management in hospitals. M. L. Brandeau, F. Sainfort, and W. P. Pierskalla (eds.), *Operations research and Health Care, International Series in Operations Research & Management Science* **70**(2) 15–41.

[101] Gross, D., D. R. Miller. 1984. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research* **32**(2) 343–361.

[102] Gross, D., J.F. Shortle, J. M. Thompson, C.M. Harris. 2008. *Fundamentals of queueing theory (4th Edition)*. Wiley Series in Probability and Statistics, Wiley-Blackwell.

[103] Gunal, M. M., M. Pidd. 2009. Understanding target-driven action in emergency department performance using simulation. *Emergency Medicine Journal* **26**(10) 724–727.

[104] Gunal, M. M., M. Pidd. 2010. Discrete event simulation for performance modelling in health care: a review of the literature. *Journal of Simulation* **4**(1) 42–51.

[105] Gurvich, I., J. Luedtke, T. Tezcan. 2010. Staffng call centers with uncertain demand forecasts: a chance-constrained optimization approach. *Management Science* **56**(7) 1093–1115.

[106] Halfin, S., W. Whitt. 1981. Heavy-traffic limits for queues with many exponential servers. *Operations Research* **29**(3) 567–588.

[107] Hampshire, R. C., O. B. Jennings, W. A. Massey. 2009. A timevarying call center design via Lagrangian mechanics. *Probability in the Engineering and Informational Sciences* **23** 231–259.

[108] Harrison, J. M., A. Zeevi. 2005. A method for staffing large call centers based on stochastic fluid models. *Manufacturing & Service Operations Management* **7**(1) 20–36.

[109] Helber, S., K. Henken. 2010. Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials. *OR Spectrum* **32**(1/4) 109–134.

[110] Henderson, S. G., A. J. Mason. 1998. Rostering by iterating integer programming and simulation. D. Medeiros, and E. Watson (eds.), *Proceedings of the 1998 Winter Simulation Conference* 677–683.

[111] Henderson, S. G., A. J. Mason, I. Ziedins, R. Thomson. 1999. A Heuristic for determining efficient staffing requirements for call centres. *Working paper*, University of Auckland, Auckland, New Zealand.

[112] Heskett, J. L., W. E. Sasser, C. W. L. Hart. 1990. *Service breakthroughs*. The Free Press, New York.

[113] Heyman, D. P., W. Whitt. 1984. The asymptotic behavior of queues with time-varying arrival rates. *Journal of Applied Probability* **21**(1) 143–156.

[114] Hojati, M., A. S. Patil. 2011. An integer linear programming-based heuristic for scheduling heterogeneous, part-time service employees. *European Journal of Operational Research* **209**(1) 37–50.

[115] Hueter, J., W. Swart. 1998. An integrated labor-management system for Taco Bell. *Interfaces* **28**(1) 75–91.

[116] Hulshof, P. J., N. Kortbeek, R. J. Boucherie, E. W. Hans, P. J. Bakker. 2012. Taxonomic classification of planning decisions in health care: a structured review of the state of the art in OR/MS. *Health Systems* **1**(2) 129–175.

[117] Hung, G. R., S. R. Whitehouse, C. B. O'Neill, A. P. Gray, N. Kissoon. 2007. Computer modeling of patient flow in a pediatric emergency department using discrete event simulation. *Pediatric Emergency Care* **23**(1) 5–10.

[118] Hur, D., V. Mabert, K. Bretthauer. 2004. Real-time work schedule adjustment decisions: an investigation and evaluation. *Production and Operations Management* **13** 322–339.

[119] Hwang, J., L. Gao, W. Jang. 2010. Joint demand and capacity management in a restaurant system. *European Journal of Operational Research* **207**(1) 465–472.

[120] Ingolfsson, A., A. Haque, A. Umnikov. 2002. Accounting for time-varying queueing effects in workforce scheduling. *European Journal of Operational Research* **139**(3) 585–597.

[121] Ingolfsson, A., E. Akhmetshina, S. Budge, Y. Li. 2007. A survey and experimental comparison of service level approximation methods for non-stationary $M_t/M/s_t$ queueing systems with exhaustive discipline. *INFORMS Journal on Computing* **19**(2) 201–214.

[122] Ingolfsson, A., F. Campello, X. Wu, E. Cabral. 2010. Combining integer programming and the randomization method to schedule employees. *European Journal of Operational Research* **202**(1) 153–163.

[123] Ingolfsson, A., 2005. Modeling the $M_t/M/s_t$ queue with an exhaustive discipline. *Working paper*, University of Alberta, Canada.

[124] Iravani, F., B. Balcioğlu. 2008. Approximations for the $M/GI/N + GI$ type call center. *Queueing Systems* **58**(2) 137–153.

[125] Izady, N., D. J. Worthington. 2012. Setting staffing requirements for time-dependent queueing networks: The case of accident and emergency departments. *European Journal of Operational Research* **219**(3) 531–540.

[126] Izady, N. 2010. On queues with time-varying demand. *PhD Dissertation*, University of Lancaster, Lancaster, UK.

[127] Jack, E. P., T. A. Bedics, C. E. McCary. 2006. Operational challenges in the call center industry: a case study and resource-based framework. *Managing Service Quality* **16**(5) 477–500.

[128] Jack, E. P., T. L. Powers. 2009. A review and synthesis of demand management, capacity management and performance in health-care services. *International Journal of Management Reviews* **11**(2) 149–174.

[129] Jacobson, S. H., S. N. Hall, J. R. Swisher. 2006. Discrete-event simulation of health care systems. *Patient flow: reducing delay in healthcare delivery*, Springer US, 211–252.

[130] Jagerman, D. L. 1975. Nonstationary blocking in telephone traffic. *Bell Syst. Tech.* **54** 625–661.

[131] Jennings, O. B., A. Mandelbaum, W. A. Massey, W. Whitt. 1996. Server staffing to meet time-varying demand. *Management Science* **42**(10) 1383–1394.

[132] Jensen, A. 1953. Markov Chains as an Aid in the Study of Markov Processes. *Skand. Aktuarietidskrift* **3** 87–91.

[133] Jiménez, T., G. Koole. 2004. Scaling and comparison of fluid limits of queues applied to call centers with time varying parameters. *OR Spectrum* **26**(3) 413–422.

[134] Johnson, M., S. Myers, J. Wineholt, M. Pollack, A.vL. Kusmiesz. 2009. Patients who leave the emergency department without being seen. *Journal of Emergency Nursing* **35**(2) 105–108.

[135] Jones, S. A., M. P Joy, J. Pearson. 2002. Forecasting demand of emergency care. *Health care management science* **5**(4) 297–305.

[136] Jones, S. S., A. Thomas, R. S. Evans, S. J. Welch, P. J. Haug, G. L. Snow. 2008. Forecasting daily patient volumes in the emergency department. *Academic Emergency Medicine* **15**(2) 159–170.

[137] Jongbloed, G., G. Koole. 2001. Managing uncertainty in call centres using Poisson mixtures. *Applied Stochastic Models in Business and Industry* **17**(4) 307–318.

[138] Jun, J. B., S. H. Jacobson, J. R. Swisher. 1999. Application of discrete-event simulation in health care clinics: A survey. *Journal of the Operational Research Society* **50**(2) 109–123.

[139] Kabak, Ö., F. Ülengin, E. Aktaş, Ş. Önsel, Y. I. Topcu. 2008. Efficient shift scheduling in the retail sector through two-stage optimization. *European Journal of Operational Research* **184**(1) 76–90.

[140] Keith, E. G. 1979. Operator scheduling. *AIII Trans* **11** 37–41.

[141] Kelley, J. E. Jr. 1960. The Cutting-Plane Method for Solving Convex Programs. *J. Soc. Indust. and Appl. Math.* **8**(4) 703–712.

[142] Kendall, D. G. 1953. Stochastic Processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. *The Annals of Mathematical Statistics* **24**(3) 338–354.

[143] Kim, J. W., S. H. Ha. 2010. Consecutive staffing solution using simulation in the contact center.*Industrial Management & Data Systems* **110**(5) 718–730.

[144] Kim, J.W., S. H. Ha. 2012. Advanced workforce management for effective customer services. *Quality & Quantity* **46**(6) 1715–1726.

[145] Kim, S. H., W. Whitt. 2013. Are call center and hospital arrivals well modeled by nonhomogeneous poisson processes? *Working paper*, Columbia University, New York, US.

[146] Kimes, S. E. 1989. Yield management: a tool for capacity-considered service firms. *Journal of Operations Management* **8**(4) 348–363.

[147] Klassen, K. J., T. R. Rohleder. 2001. Combining operations and marketing to manage capacity and demand in services. *Service Industries Journal* **21**(2) 1–30.

[148] Klassen, K. J., T. R. Rohleder. 2002. Demand and capacity management decisions in services: how they impact on one another. *International Journal of Operations & Production Management* **22**(5) 527–548.

[149] Klassen, K. J., T. R. Rohleder. 2004. Using customer motivations to reduce peak demand: does it work? *The Service Industries Journal* **24**(5) 53–69.

[150] Kolesar, P. J., K. L. Rider, T. B. Crabill, W. E. Walker. 1975. A queuing-linear programming approach to scheduling police patrol cars. *Operations Research* **23**(6) 1045–1062.

[151] Komarudin, M.-A. Guerry, T. De Feyter, G. Vanden Berghe. 2013. The roster quality staffing problem – A methodology for improving the roster quality by modifying the personnel structure.*European Journal of Operational Research*, To appear.

[152] Koole, G., A. Mandelbaum. 2002. Queueing models of call centers: an introduction. *Annals of Operations Research* **113**(1) 41–59.

[153] Koole, G., A. Pot. 2006. An overview of routing and staffing algorithms in multi-skill customer contact centers. *Working paper*, VU University Amsterdam, the Netherlands.

[154] Koole, G., R. Righter. 1998. Optimal control of tandem reentrant queues. *Queueing Systems- Theory and Applications* **28**(4) 337–347.

[155] Koole, G., E. van der Sluis. 2003. Optimal shift scheduling with a global service level constraint. *IIE Transactions* **35**(11) 1049–1055.

[156] Koopman, B. O. 1972. Air-terminal queues under time-dependent conditions. *Operations Research* **20**(6) 1089–1114.

[157] Lam, S., M. Vandenbosch, M. Pearce. 1998. Retail sales force scheduling based on store traffic forecasting. *Journal of Retailing* **74**(1) 61–88.

[158] Law, A. M., W. D. Kelton. 2000. *Simulation modeling and analysis.* McGraw-Hill series in industrial engineering and management science, McGraw-Hill, Boston, US.

[159] Le Minh, D. 1978. The discrete-time single-server queue with time-inhomogeneous compound Poisson input and general service time distribution. *Journal of Applied Probability* 590–601.

[160] Le, L. T. 2006. Demand management at congested airports: How far are we from utopia? *PhD dissertation*, George Mason University.

[161] Liao, S., G. Koole, C. van Delft, O. Jouini. 2012. Staffing a call center with uncertain non-stationary arrival rate and flexibility. *OR Spectrum* **34** 691–721.

[162] Liao, S., C. van Delft, J. P. Vial. 2013. Distributionally robust workforce scheduling in call centres with uncertain arrival rates. *Optimization Methods and Software* **28**(3) 501–522.

[163] Lim, M. E., A. Worster, R. Goeree, J. E. Tarride. 2013. Simulating an emergency department: the importance of modeling the interactions between physicians and delegates in a discrete event simulation. *BMC medical informatics and decision making* **13**(1).

[164] Lin, C. K. Y., K. F. Lai, S. L. Hung. 2000. Development of a workforce management system for a customer hotline service.*Computers & Operations Research* **27**(10) 987–1004.

[165] Littler, R. A., D. Whitaker 1997. Estimating staffing requirements at an airport terminal. *Journal of the Operational Research Society* **48**(2) 124–131.

[166] Liu, Y., W. Whitt. 2011. A network of time-varying many-server fluid queues with customer abandonment. *Operations Research* **59**(4) 835–846.

[167] Liu, Y., W. Whitt. 2011. Large-time asymptotics for the $G_t/M_t/s_t + GI_t$ many-server fluid queue with abandonment. *Queueing systems* **67**(2) 145–182.

[168] Liu, Y., W. Whitt. 2012. The $G_t/GI/s_t+GI$ many-server fluid queue. *Queueing Systems* **71**(4) 405–444.

[169] Liu, Y., W. Whitt. 2012. A many-server fluid limit for the $G_t/GI/s_t + GI$ queueing model experiencing periods of overloading. *OR Letters* **40** 307–312.

[170] Liu, Y., W. Whitt. 2009. Stabilizing customer abandonment in many-server queues with time-varying arrivals. *Working paper*, Columbia University, New York, US.

[171] Liu, Y., W. Whitt. 2010. A Fluid Approximation for the $GI_t/GI/s_t + GI$ Queue. *Working paper*, Columbia University, New York, US.

[172] Liu, Y., W. Whitt. 2013. Algorithms for time-varying networks of many-server fluid queues. *INFORMS Journal on Computing*. To appear.

[173] Maman S. 2009. Uncertainty in the demand for service: the case of call centers and emergency departments. *M.Sc. Thesis*, Technion, Israel Institute of Technology, Israel.

[174] Mandelbaum, A., P. Momğilović. 2008. Queues with many servers: The virtual waiting-time process in the QED regime. *Mathematics of Operations Research* **33**(3) 561–586.

[175] Mandelbaum, A., P. Momğilović. 2012. Queues with many servers and impatient customers. *Mathematics of Operations Research* **37**(1) 41–65.

[176] Mandelbaum, A., W. A. Massey. 1995. Strong approximations for time-dependent queues. *Mathematics of Operations Research* **20**(1) 33–64.

[177] Mandelbaum, A., W. Massey, M. Reiman. 1998. Strong approximations for Markovian service networks. *Queueing Systems* **30**(1) 149–201.

[178] Mandelbaum, A., W. A. Massey, M. I. Reiman, R. Rider. 1999. Time varying multiserver queues with abandonments and retrials. *Proceedings of the 16th International Teletraffic Conference* **3** 355–364.

[179] Mandelbaum, A., W. A. Massey, M. I. Reiman, A. Stolyar. 1999. Waiting time asymptotics for time varying multiserver queues with abandonment and retrials. *Proc. 37th Allerton Conf. Monticello*, IL, 1095–1104.

[180] Mandelbaum, A., W. A. Massey. M. I. Reiman, A. Stolyar, B. Rider. 2002. Queue lengths and waiting times for multiserver queues with abandonment and retrials. *Telecommunication Systems* **21**(2-4) 149–171.

[181] Mandelbaum, A. 2003. Call centers (centres): Research bibliography with abstracts. Version 3. Available online at http://iew3.technion.ac.il/Labs/Serveng/

[182] Mason, A. J., D. M. Ryan, D. M. Panton. 1998. Integrated Simulation, Heuristic and Optimisation Approaches to Staff Scheduling. *Operations Research* **46**(2) 161–175.

[183] Mason, S., E. J. Weber, J. Coster, J. Freeman, T. Locker. 2012. Time patients spend in the emergency department: England's 4-hour rulea case of hitting the target but missing the point? *Annals of emergency medicine* **59**(5) 341–349.

[184] Massey, W. A., W. Whitt. 1994. An analysis of the modified offered-load approximation for the nonstationary erlang loss model. *The Annals of Applied Probability* **4**(4) 1145–1160.

[185] Massey, W. A., W. Whitt. 1997. Peak congestion in multi-server service systems with slowly varying arrival rates. *Queueing Systems* **25**(1) 157–172.

[186] Matteson, D. S., M. W. McLean, D. B. Woodard, S. G. Henderson. 2011. Forecasting emergency medical service call arrival rates. *Annals of Applied Statistics* **5**(2B) 1379–1406.

[187] McGuire, F. 1994. Using simulation to reduce length of stay in emergency departments. M. S. Manivannan, J. D. Tew (eds.), *Proceedings of the 26th conference on Winter simulation (WSC '94)*, Society for Computer Simulation International, San Diego, CA, USA, 861–867.

[188] Mehrotra, V., J. Fama. 2003. Call center simulation modeling: methods, challenges and opportunities. S. Chick, P. J. Snchez, D. Ferrin, and D. J. Morrice (eds.), *Proceedings of the 2003 Winter Simulation Conference* 135–143.

[189] Mehrotra, V., O. Ozluk, R. Saltzman. 2010. Intelligent procedures for intra-day updating of call center agent schedules. *Production and Operations Management* **19**(3) 353–367.

[190] Mellor, E. F. 1986. Shift work and flexitime: How prevalent are they? *Monthly Lab. Rev.* **109**(14).

[191] Millán-Ruiz, D., J. I. Hidalgo. 2013. Forecasting call centre arrivals. *Journal of Forecasting* **32**(7) 628–638.

[192] Morzuch, B. J., Allen, P. G. 2006. Forecasting hospital emergency department arrivals. *26th Annual Symposium on Forecasting*, Santander, Spain.

[193] Nah, J. E., S. Kim. 2013. Workforce planning and deployment for a hospital reservation call center with abandonment cost and multiple tasks. *Computers & Industrial Engineering* 65(2) 297–309.

[194] Netessine, S., M. L. Fisher, J. Krishnan, 2010. Labor planning, execution, and retail store performance: An exploratory investigation. *Working paper*, University of Pennsylvania, US.

[195] Paul, S. A., M. C. Reddy, C. J. DeFlitch. 2010. A systematic review of simulation studies investigating emergency department overcrowding. *Simulation* **86**(8-9) 559–571.

[196] Pendergraft, D. R., C. V. Robertson, S. Shrader. 2004. Simulation of an airport passenger security system. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, (eds.), *Proceedings of the 36th conference on Winter simulation* 874–878.

[197] Pitt, M. 1997. A generalised simulation system to support strategic resource planning in healthcare. S. Andradottir, K. J. Healy, D. H. Withers, B. L. Nelson (eds.) *Proceedings of the 29th conference on Winter simulation* IEEE Computer Society, Washington, DC, USA, 1155–1162.

[198] Pot, A., S. Bhulai, G. Koole. 2008. A simple staffing method for multiskill call centers. *Manufacturing & Service Operations Management* **10**(3) 421–428.

[199] Pullman, M., S. Rodgers. 2010. Capacity management for hospitality and tourism: A review of current approaches. *International Journal of Hospitality Management* **29**(1) 177–187.

[200] Quinn, P., B. Andrews, H. Parsons. 1991. Allocating telecommunications resources at L. L. Bean, Inc. *Interfaces* **21**(1) 75-91.

[201] Rekik, M., J.-F. Cordeau, F. Soumis. 2010. Implicit shift scheduling with multiple breaks and work stretch duration restrictions. *Journal of Scheduling* **13**(1) 49–75.

[202] Ridley, A. D., M. C. Fu, W. A. Massey. 2003. Customer relations management: call center operations: Fluid approximations for a priority call center with time-varying arrivals. *Proceedings of the 35th Conference on Winter Simulation*, New Orleans, LA, 2, 1817–1823.

[203] Robbins, T. R., T. P. Harrison. 2010. A stochastic programming model for scheduling call centers with global service level agreements. *European Journal of Operational Research* **207** 1608–1617.

[204] Robbins, T. R., D. J. Medeiros, P. Dum. 2006. Evaluating arrival rate uncertainty in call centers. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto (eds.), *Proceedings of the 2006 Winter Simulation Conference* 2180–2187.

[205] Robbins, T. R. 2007. Managing service capacity under uncertainty. *Ph.D. thesis*, Penn State University.

[206] Robertson, C. V., S. Shrader, D. R. Pendergraft, L. M. Johnson, K. S. Silbert. 2002. The role of modeling demand in process re-engineering. *Proceedings of the 2002 Winter Simulation Conference* 1454–1458.

[207] Rothkopf, M. H., S. S. Oren. 1979. A closure approximation for the nonstationary $M/M/s$ Queue. *Management Science* **25**(6) 522–534.

[208] Roubos, A., S. Bhulai, G. Koole. 2011. Flexible staffing for call centers with non-stationary arrival rates. *Working paper*, VU University Amsterdam, the Netherlands.

[209] Saccani, N. 2012. Forecasting for capacity management in call centres: combining methods, organization, people and technology. *IMA Journal of Management Mathematics Advance Access*.

[210] Saltzman, R., V. Mehrotra. 2007. Managing trade-offs in call center agent scheduling: methodology and case study. *Proceedings of the 2007 summer computer simulation conference*, Society for Computer Simulation International 643–651.

[211] Saltzman, R. M. 2005. A hybrid approach to minimize the cost of staffing a call center. *International Journal of Operations and Quantitative Management* 11(1) 1–14.

[212] Sasser, W. E. 1976. Match supply and demand in service industries. *Harvard Business Review* **54**(6) 133–140.

[213] Shampine, L. F., M.W. Reichelt. 1997. The MATLAB ODE Suite. *SIAM Journal on Scientific Computing* **18**(1) 1–22.

[214] Shen, H., & Huang, J. Z. (2008). Forecasting time series of inhomogeneous Poisson processes with application to call center workforce management. *The Annals of Applied Statistics* 601–623.

[215] Showalter, M. J., J. D. White. 1991. An integrated model for demand-output management in service organisations: implications for future research. *International Journal of Operations & Production Management* **11**(1) 51–67.

[216] Siferd, S. P., W. C. Benton, L. P. Ritzman. 1992. Strategies for service systems. *European Journal of Operational Research* **56**(3) 291–303.

[217] Sill, B.T. 1991. Capacity management: Making your service delivery more productive, *The Cornell Hotel and Restaurant Administration Quarterly* **31**(4) 76–87.

[218] Sinreich, D., O. Jabali. 2007. Staggered work shifts: a way to downsize and restructure an emergency department workforce yet maintain current operational performance. *Health Care Management Science* **10** 293–308.

[219] Sinreich, D., Y. N. Marmor. 2004. A simple and intuitive simulation tool for analyzing emergency department operations. *Proceedings of the 36th conference on Winter simulation (WSC '04)* 1994–2002.

[220] Steckley, S. G., S. G. Henderson, V. Mehrotra. 2004. Service system planning in the presence of random arrival rate. *Working paper*, Cornell University.

[221] Steckley, S. G., S. G. Henderson, and V. Mehrotra. 2009. Forecast errors in service systems. *Probability in the Engineering and Informational Sciences* **23**(2) 305–332.

[222] Stolletz, R., S. Lagershausen. 2013. Time-dependent performance evaluation for loss-waiting queues with arbitrary distributions. *International Journal of Production Research* **51**(5) 1366–1378.

[223] Stolletz, R. 2008a. Approximation of the non-stationary $M_t/M_t/c_t$-queue using stationary queueing models: The stationary backlog-carryover approach. *European Journal of Operational Research* **190** 478–493.

[224] Stolletz, R. 2008b. Non-stationary delay analysis of runway systems. *OR Spectrum* **30**(1) 191–213.

[225] Stolletz, R. 2011. Analysis of passenger queues at airport terminals. *Research in Transportation Business & Management* **1**(1) 144–149.

[226] Taaffe, M., K. Ong. 1987. Approximating nonstationary $Ph(t)/Ph(t)/l/c$ queueing systems. *Annals of Operations Research* **8**(1) 103–116.

[227] Takakuwa, S., H. Shiozaki. 2004. Functional analysis for operating emergency department of a general hospital. *Proceedings of the 36th conference on Winter simulation(WSC '04)* 2003–2011.

[228] Taylor, J. W. 2008. A comparison of univariate time series methods for forecasting intraday arrivals at a call center. *Management Science* **54**(2) 253–265.

[229] Testik, M. C., J. K. Cochran, G. C. Runger. 2004. Adaptive server staffing in the presence of time-varying arrivals: a feed-forward control approach. *The Journal of the Operational Research Society* **55**(3) 233–239.

[230] Thompson, G. M. 1993. Accounting for the multi-period impact of service when determining employee requirements for labor scheduling. *Journal of Operations Management* **11**(3) 269–287.

[231] Thompson, G.M. 1995. Labor scheduling using NPV estimates of the marginal benefit of additional labor capacity. *Journal of Operations Management* **13**(1) 67–86.

[232] Thompson, G. M. 1997. Labor staffing and scheduling models for controlling service levels. *Naval Research Logistics* **44**(8) 719–740.

[233] Thompson, G. M. 1998a. Labor scheduling, Part 1: Forecasting demand. *The Cornell Hotel and Restaurant Administration Quarterly* **39**(5) 22–31.

[234] Thompson, G. M. 1998b. Labor scheduling, Part 2. *The Cornell Hotel and Restaurant Administration Quarterly* **39**(6) 26–37.

[235] Thompson, G. M. 1999. Labor scheduling, Part 4: Controlling Workforce Schedules in Real Time. *The Cornell Hotel and Restaurant Administration Quarterly* **40**(3) 85–96.

[236] Thompson, G. M. 1999a. Labor scheduling, Part 3: Developing a workforce schedule. *The Cornell Hotel and Restaurant Administration Quarterly* **40**(1) 86–94.

[237] Thompson, G. M. 1999b. Labor scheduling, Part 4 Controlling workforce schedules in real time. *The Cornell Hotel and Restaurant Administration Quarterly* **40**(3) 85–96.

[238] Tsai, W. K., P. E. Cantrell. 1989. A simple derivation of transient queue statistics and applications. *Performance evaluation* 10(2) 103–114.

[239] Tulkens, H. 1993. On FDH analysis: some methodological issues and applications to retail banking, courts and urban transit. *Journal of Productivity Analysis* **4** 183–210.

[240] Van den Bergh, J., J. Beliën, P. De Bruecker, E. Demeulemeester, L. De Boeck. 2013. Personnel scheduling: A literature review. *European Journal of Operational Research* **226**(3) 367–385.

[241] Vile, J. 2013. Time-dependent stochastic modelling for predicting demand and scheduling of emergency medical services. *PhD dissertation*, Cardiff University, UK.

[242] Wall, A. D., D. J. Worthington. 1994. Using Discrete Distributions to Approximate General Service Time Distributions in Queueing Models. *The Journal of the Operational Research Society* **45**(12) 1398–1404.

[243] Wall, A. D., D. J. Worthington. 2007. Time-dependent analysis of virtual waiting time behaviour in discrete time queues. *European Journal of Operational Research* **178**(2) 482–499.

[244] Wargon, M., B. Guidet, T. D. Hoang, G. Hejblum. 2009. A systematic review of models for forecasting the number of emergency department visits. *Emergency Medicine Journal* **26**(6) 395–399.

[245] White P. K. Jr. 2005. A survey of data resources for simulating patient flows in healthcare delivery systems. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines (eds.), *Proceedings of the 2005 Winter Simulation Conference* 926–935.

[246] Whitt, W. 1991. The pointwise stationary approximation for $M_t/M_t/s$ queues is asymptotically correct as the rates increase. *Management Science* **37**(3) 307–314.

[247] Whitt, W. 1992. Understanding the efficiency of multi-server service systems. *Management Science* **38**(5) 708–723.

[248] Whitt, W. 1999. Dynamic staffng in a telephone call center aiming to immediately answer all calls. *Operations Research Letters* **24** 205–212.

[249] Whitt, W. 2005. Engineering solution of a basic call-center model. *Management Science* **51**(2) 221–235.

[250] Whitt, W. 2006. Staffing a call center with uncertain arrival rate and absenteeism. *Production and Operations Management* **15**(1) 88–102.

[251] Whitt, W. 2006. Fluid models for multiserver queues with abandonments. *Operations Research* **54**(1) 37–54.

[252] Whitt, W. 2006c. Staffing a call center with uncertain arrival rate and absenteeism. *Production and Operations Management* **15**(1) 88–102.

[253] Whitt, W. 2007. What you should know about queueing models to set staffing requirements in service systems. *Naval Research Logistics* **54**(5) 476–484.

[254] Whitt, W. 2013. OM Forum – Offered load analysis for staffing. *Manufacturing & Service Operations Management* **15**(2) 166–169.

[255] Yom-Tov, G. 2010. Queues in Hospitals: Queueing networks with reentering customers in the QED regime. *Ph.D. Thesis*, Technion - Israel Institute of Technology, Haifa, Israel.

[256] Zeithaml, V. A., A. Parasuraman, L. L. Berry. 1985. Problems and strategies in services marketing. *Journal of Marketing* **49** 33–46.

[257] Zeltyn, S., A. Mandelbaum. 2005. Call Centers with Impatient Customers: Many-Server Asymptotics of the $M/M/n+G$ Queue. *Queueing Syst. Theory Appl.* **51**(3-4) 361–402.

[258] Zeltyn, S., Y. N. Marmor, A. Mandelbaum, B. Carmeli, O. Greenshpan, Y. Mesika, S. Wasserkrug, P. Vortman, A. Shtub, T. Lauterman, D. Schwartz, K. Moskovitch, S. Tzafrir, F. Basis. 2011. Simulation-based models of emergency departments: Operational, tactical, and strategic staffing. *ACM Transactions on Modeling and Computer Simulation* **21**(4) 1–24.

[259] Zhang, Y., M. L. Puterman, M. Nelson, D. Atkins. 2012. A simulation optimization approach to long-term care capacity planning. *Operations Research* **60**(2) 249–261.

Doctoral Dissertations from the Faculty of Business and Economics

Doctoral dissertations from the Faculty of Business and Economics, see:
http://www.kuleuven.ac.be/doctoraatsverdediging/archief.htm.